

Technische Universität Wien

DIPLOMARBEIT

**Steuerprogramm für ein Infrarot-Informations- und  
Orientierungssystem für Blinde und Sehbehinderte**

ausgeführt am Institut für  
Allgemeine Elektrotechnik und Elektronik  
der Technischen Universität Wien

unter Anleitung von

Ao. Univ.-Prof. Dipl.-Ing. Dr. techn. Franz Zach  
und  
Dipl.-Ing Peter Mayer

durch

Christian Hintersteiner  
Matr.Nr. 8725676  
Wasserring 2, 3300 Winklarn

Winklarn, im August 1993

# Inhaltsverzeichnis

1. Einleitung .....	1
2. Aufgabenstellung .....	1
3. Systembeschreibung .....	2
3.1 Allgemein .....	2
3.2 Hardware .....	3
3.3 Infrarot-Datenprotokoll .....	3
3.4 Software-Datenprotokoll .....	4
3.5 Informationskanäle .....	8
3.5.1 Sicherheit .....	9
3.5.2 Verkehrsampel .....	9
3.5.3 Orientierung .....	10
3.5.4 Verkehr .....	10
3.5.5 Versorgung .....	11
3.5.6 Touristik/Museum/Ausstellung .....	11
3.5.7 Datenschnittstellen .....	11
3.6 Benutzerschnittstelle .....	12
3.6.1 Der Drehschalter .....	12
3.6.2 Der Taster .....	12
3.7 Programmauswahl .....	14
3.7.1 Informationskanäle 0-6 .....	14
3.7.2 Einkauf .....	14
3.7.3 Freizeit .....	15
3.7.4 Fortbewegung in Gebäuden .....	15
3.7.5 Orientieren in neuer Umgebung .....	15
3.7.6 Sanitäre Einrichtungen .....	15
3.7.7 Hilfe .....	15
3.7.8 Parameter einstellen .....	16

4. Software	18
4.1 Programmstruktur	18
4.2 Speicherverwaltung	19
4.3 Empfangen	22
4.4 Speichern und Verwalten	22
4.5 Auswahl der zu sprechenden Meldungen	23
4.6 Sprechen	24
4.7 Auswerten der Bedienelemente	25
4.8 Funktionsaufrufe	26
5. Bedienungsanleitung	27
5.1 Sonderfunktionen	28
5.2 Die Bedienelemente	30
6. Programmbeschreibung	32
6.1 Allgemeines	32
6.2 Systemvariablen	33
6.3 Buffer	36
6.4 IRIS.C	39
6.5 RXD.C	39
6.6 SPEAK.C	39
6.7 USER.C	40
6.8 TIME.C	40
Anhang A Codetabellen	43
A1 Infogrammcodes	43
A2 Lautschriftcodes	46
A3 Kurzinformation	48

	III
Anhang B Flußdiagramme .....	49
B1 main .....	49
B2 dlt_administration .....	50
B3 check_kreisw .....	51
B4 choose_message_to_speak .....	52
B5 speak_msg .....	53
B6 chk_key .....	54
B7 chk_swf .....	55
Literaturverzeichnis .....	56

# 1. Einleitung

Am Institut für Allgemeine Elektrotechnik und Elektronik der TU Wien wurde von der Arbeitsgruppe für Rehabilitationstechnik ein **InfraRot-Informationen- und OrientierungsSystem** für Blinde und Sehbehinderte, genannt "I.R.I.S.", im Auftrag der Stadt Wien (MA46) entwickelt (/MAY92/), und soll in Zukunft dieser Personengruppe eine Mobilitätsverbesserung im Nahverkehr ermöglichen. Das System besteht aus fix installierten Infrarotsendern und Miniaturempfängern, welche von den Blinden getragen werden. Die grundsätzliche Aufgabe des Empfängers ist die Umsetzung der empfangenen Daten in synthetische Sprache. Durch ein spezielles Codierungsverfahren kann das System mehrsprachig eingesetzt werden, sodaß erst im Empfänger bestimmt wird, in welcher Sprache die Information weitergegeben wird. Durch die Möglichkeit Phoneme zu übertragen, wird auch die richtige Aussprache von Eigennamen möglich.

## 2. Aufgabenstellung

Die Hauptaufgabe des Systems ist, wie bereits in der Einleitung erwähnt, eine Mobilitätsverbesserung für Blinde und Sehbehinderte zu erreichen. Zu diesem Zweck reicht jedoch eine einfache Sprachumsetzung der empfangenen Daten in synthetische Sprache nicht aus, vielmehr ist es notwendig eine Unterscheidung in Informationsarten zu treffen und eine selektive Auswahl derselben anzubieten. Zu diesem Zweck sind zwei Bedienelemente, nämlich ein Taster und ein Drehschalter, am Empfänger angebracht. Die Einschränkung auf wenige Bedienelemente wurde bewußt in Kauf genommen, um einerseits eine einfache Bedienung zu ermöglichen, und andererseits den Platzbedarf am Empfänger gering zu halten.

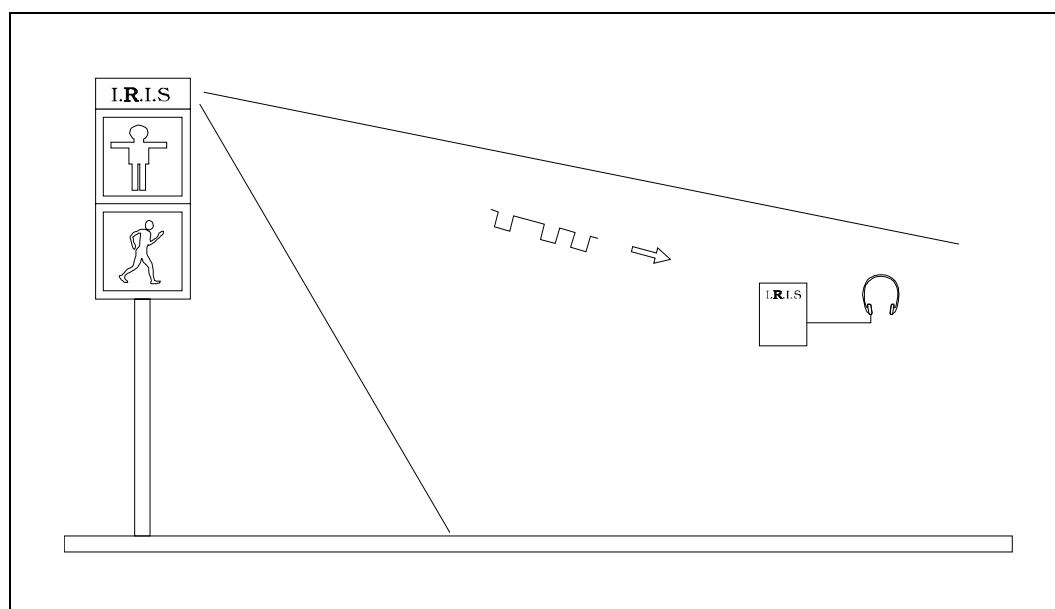
Ziel dieser Diplomarbeit ist es, ein Steuerprogramm für den Empfänger zu entwickeln, das die empfangenen Informationen einerseits aufbereitet, speichert und verwaltet, und andererseits dem Benutzer in sinnvoller Weise anbietet.

Es ist dabei von besonderer Bedeutung, daß die Benutzung möglichst weniger Interaktionen bedarf, da sonst die Aufmerksamkeit von der Umwelt abgelenkt wird, und es eher zu einer Verschlechterung der Mobilität kommen würde. Um den Bedürfnissen verschiedener Benutzertypen gerecht werden zu können, soll eine Anpassung des Systems durch den Benutzer ermöglicht werden.

## 3. Systembeschreibung

### 3.1 Allgemein

Das System besteht aus den zwei grundsätzlichen Komponenten, Sender und Empfänger. Die Sender sind an neuralgischen Punkten wie Kreuzungen, Ampeln, Haltestellen, Verkehrsmitteln, etc. angebracht und senden die Informationen in codierter Form im Infrarotbereich aus.



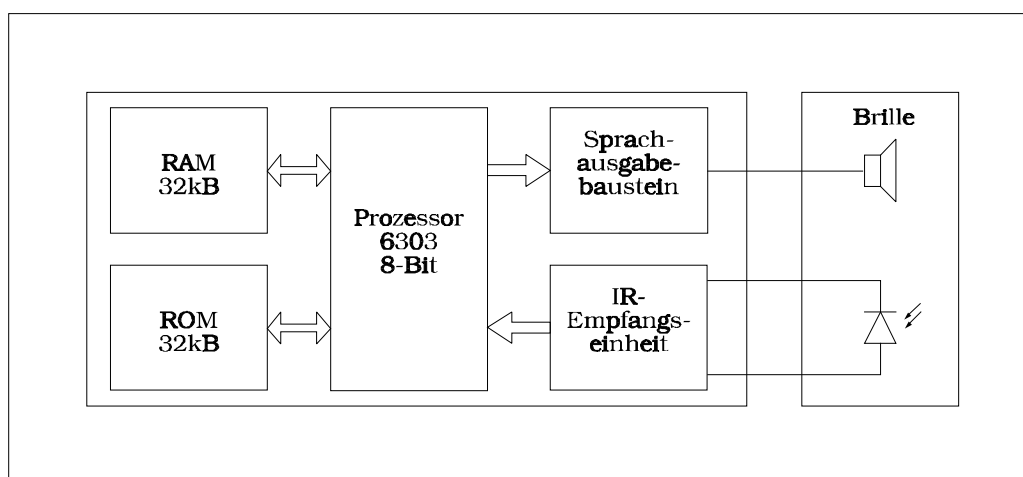
**Abb. 1** Typische Anbringung eines Senders

Der Empfänger kann in unterschiedlichen Formen hergestellt werden. Der Prototyp besteht aus einer Brille, in der die Empfangsdioden samt Linsen, die IR-Empfangselektronik und die Kopfhörer untergebracht sind, sowie einem Gehäuse, das die Digitalelektronik beinhaltet und an dem die Bedienelemente angebracht sind. Dieses Gehäuse kann wie ein Walkman am Gürtel befestigt werden. Die Elektronik besteht aus einem Codewandler, einem Sprachsynthesemodul und einem Mikroprozessor, der das Herzstück darstellt.

## 3.2 Hardware

Die Sendestufe des Senders und die Infrarotempfängseinheit des Empfängers entstammen der Diplomarbeit von Helmut KAUER (/KAU92/).

Das Mikroprozessorsystem wurde im von der Gemeinde Wien geförderten Projekt "IRIS" entwickelt. Die Schaltpläne können dem dazugehörigen Projektbericht (/MAY92/) entnommen werden.



**Abb. 2** Blockschaltbild des Empfängers

## 3.3 Infrarot-Datenprotokoll

Die PCM-Übertragungsstrecke wurde im Rahmen der Diplomarbeit von Helmut Kauer entwickelt, der ein serielles Übertragungsprotokoll in Manchesterkodierung wählte, das sich durch hohe Datensicherheit und Übertragungsgeschwindigkeit (31.25 kBit/s) auszeichnet.

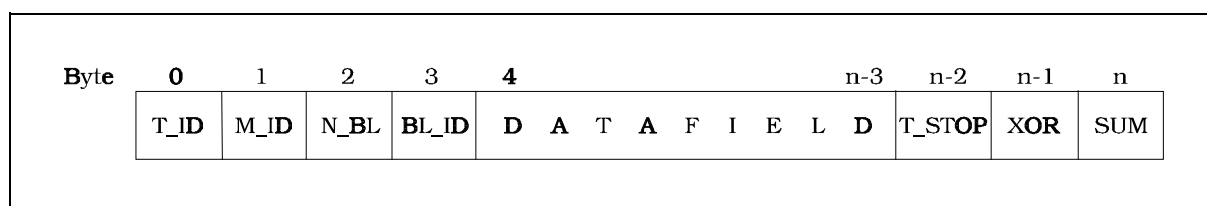
Das Protokoll - es entspricht dem IBM® 5250 Standard - unterscheidet 7 verschiedene Kanäle, welchen die im Kapitel 3.5 definierten Informationskanäle zugeordnet werden.

Technische Daten bzw. Schaltungsentwürfe sind der Diplomarbeit von Helmut KAUER (/KAU92/) zu entnehmen.

### 3.4 Software-Datenprotokoll

Die von der Infrarotempfangseinheit erhaltenen Meldungen sind in Blöcke gegliedert. Eine Meldung besteht aus mindestens einem und höchstens 255 Blöcken. Ein Block darf, inklusive Kennungen und Prüfbyte, maximal 255 Byte lang sein.

Ein Block setzt sich folgendermaßen zusammen:



T\_ID ..... Senderidentifizierung

M\_ID ..... Meldungsidentifizierung

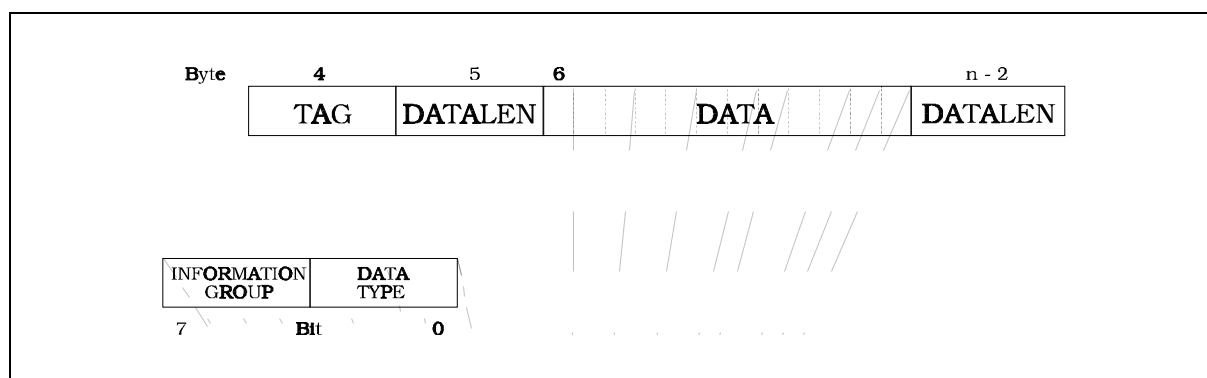
N\_BL ..... Anzahl der Blöcke aus denen die Meldung besteht

BL\_ID..... Blockidentifizierung

XOR ..... Prüfbyte (Exklusiv-Oder )

SUM ..... Prüfbyte (negative Summe)

Das DATAFIELD setzt sich aus einem oder auch mehreren Feldern der in der folgenden Abbildung gezeigten Struktur zusammen :





Das TAG-Byte ist in zwei 4Bit-Felder unterteilt.

Die Bits 0-3 bestimmen, welchem Datentyp die nach dem TAG ankommenden Daten zuzuordnen sind.

Die Bits 4-7 haben die Aufgabe die, nach dem TAG gesendeten Daten einer bestimmten Informationsgruppe zuzuordnen. Da die Informationsgruppen je nach Informationskanal unterschiedlich sind, werden sie gemeinsam im nächsten Kapitel behandelt.

### 3.4.1 Datentypen

Man unterscheidet folgende Datentypen:

Bit 3	Bit 2	Bit 1	Bit 0	Datatype	DATA
0	0	0	1	T_INFO	INFOID ~INFOID
0	0	1	0	T_PHONEM	n PHONEMBYTES
0	1	0	0	T_ASCII	n ASCIIBYTES
0	1	1	1	T_NUMBER	2 NUMBERBYTES
1	0	0	0	T_ORDINAL	2 NUMBERBYTES
1	0	1	1	T_CONTROL	2 NUMBERBYTES
1	1	0	1	T_STOP	

~INFOID ... Komplement von INFOID

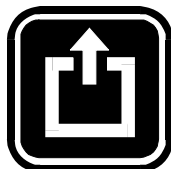
Aus der Tabelle erkennt man, daß die Datentypen T\_INFO, T\_NUMBER, T\_ORDINAL und T\_CONTROL konstante Länge besitzen.

Die Codierung der Datentypen wurde so gewählt, daß bei "Umfallen" eines Bits kein anderer Datentyp entstehen kann, und Fehlinterpretationen damit praktisch vermieden werden.

## **T\_INFO**

'I.R.I.S' ist als vielsprachiges System konzipiert. Für sehende Verkehrsteilnehmer wurden zahlreiche Pictogramme entwickelt, anhand derer sie sich auch ohne Kenntnis der Landessprache orientieren können.

Das Pictogramm



bedeutet z.B. "Der Ausgang befindet sich geradeaus" - unabhängig von einer bestimmten Sprache.

In Analogie zu dieser Einrichtung werden hier codierte Signale verwendet, die als 'Infogramme' bezeichnet werden. Ein Infogramm ist eine Informationseinheit, die zunächst nur eine bestimmte Bedeutung hat, ohne sprachlich festgelegt zu sein. Der Bedeutung "Ausgang befindet sich geradeaus" wird hier als Infogramm eine Kennziffer zugewiesen (Infogrammcodes siehe Anhang A). Wird ein solches Infogramm empfangen so wird es in der eingestellten Landessprache vom Sprachsynthesizer gesprochen, d.h. der Empfänger und nicht der Sender bestimmt die Sprache.

## **T\_PHONEM**

Kommen innerhalb eines, z.B. aus Infogrammen zusammengestellten, Textes Eigennamen vor, werden diese in Form von Lautschriftsymbolen übertragen, wodurch auch eine korrekte Aussprache gewährleistet ist.

Eine Codetabelle der derzeit verwendeten Lautschriftsymbole - eine Internationale Norm existiert noch nicht - befindet sich im Anhang A.

## **T\_ASCII**

Kann ein Text nicht vollständig aus Infogrammen gebildet werden und ist er für die Übertragung in Lautschriftsymbolen zu lang kann man ihn in Form von ASCII-Zeichen übertragen, verliert dabei jedoch den Vorteil der Sprachunabhängigkeit. Ein ASCII-Block ist eine Folge von einem oder mehreren Bytes, die direkt einem Zeichen nach dem ISO-PC8 Code entsprechen. Die Zeichenfolge wird mit einem NULL-Byte abgeschlossen.

**T\_NUMBER**

Dieser Datentyp dient zur Übertragung von Zahlen (Wertebereich 0 - 65535). Nach ihm werden zwei Bytes übertragen, wobei das höherwertige zuerst gesendet wird.

**T\_ORDINAL**

Derzeit nicht verwendet.

**T\_CONTROL**

Derzeit nicht verwendet.

**T\_STOP**

Der STOP\_TAG kennzeichnet das Ende eines Blocks. Er gehört im Prinzip zu den Datenfeldern wurde jedoch bei der Definition eines Blocks wie ein eigenständiges Steuerbyte behandelt um seine Wichtigkeit zu unterstreichen.

### 3.5 Informationskanäle

Wenn man als Sehender einmal darauf achtet, welche Informationen man akustisch und, vor allem, optisch wahrnimmt und verarbeitet, bzw. sich als nützlich merkt, so erkennt man die Notwendigkeit eines Systems, das diese Informationen auch Blinden zugänglich macht. Des Weiteren erkennt man die Unterschiedlichkeit der Informationen und die differenzierte Auswahl, die man in bestimmten Situationen trifft. Dies macht deutlich, daß eine Klassifikation bzw. Einteilung der Information zur optimalen Aufbereitung unbedingt erforderlich ist. Wie bereits in Kapitel 3.3 erwähnt, unterstützt das Infrarot-Sendeprotokoll eine Unterteilung in 7 Kanäle, welchen folgende Informationsarten zugeordnet werden:

- Sicherheit
- Verkehrsampel
- Orientierung
- Verkehr
- Versorgung
- Touristik/Kultur
- Datenschnittstelle

Da die einzelnen Informationskanäle zum Teil mehrere verschiedene Informationen in sich vereinen, bzw. sich aus Teilinformationen zusammensetzen, werden diese teilweise noch in Informationsgruppen unterteilt, um in der Lage zu sein, dem Benutzer nur die wirklich gewünschte Information mitzuteilen, bzw. Information in gekürzter Form mitzuteilen.

Der Code zur Identifizierung der Informationsgruppe wird, wie im vorhergehenden Kapitel gezeigt, im TAG-Byte (Bits 4-7) gesendet. Es wird dabei der gleiche Code wie zur Unterscheidung der Datentypen verwendet, d.h. es können maximal 7 Informationsgruppen pro Informationskanal unterschieden werden.

### 3.5.1 Sicherheit

Dieser Kanal wird primär zur Übertragung von Alarmmeldungen und Warnhinweisen mit besonderer Priorität, welche in jedem Betriebszustand dem Benutzer mitzuteilen sind, verwendet. Zur Unterscheidung von zusätzlicher, in diesem Kanal gesendeter, Information werden folgende Informationsgruppen eingeführt:

Bit 7	Bit 6	Bit 5	Bit 4	Informationsgruppe
0	0	0	1	Warnung
0	0	1	0	Erste Hilfe
0	1	0	0	Polizei

- Warnung

Tag, mit dem alle Warnhinweise mit besonderer Priorität zu senden sind.

- Erste Hilfe

Tag, mit dem der Weg zum nächstgelegenen Arzt bzw. Spital zu senden ist.

- Polizei

Tag, mit dem der Weg zur nächstgelegenen Polizeistation zu senden ist.

### 3.5.2 Verkehrsampel

Dieser Kanal dient zur Übertragung von Informationen, die zur sicheren Überquerung einer Kreuzung notwendig sind. Diese sollen über die Gehrichtung (Straßenname, Wegbeschreibung, Hinweise auf besondere Gefahren) und natürlich den Ampelstand informieren. Es ist dabei zu beachten, daß 'Grün' und 'Rot' als unterschiedliche Meldungen zu betrachten sind und daher mit unterschiedlicher Meldungsidentifizierung (M\_ID) zu senden sind.

### 3.5.3 Orientierung

Dient zur Positionsbestimmung und zur Wegbeschreibung, zu und innerhalb von Gebäuden. Zur Unterscheidung stehen folgende Informationsgruppen zur Verfügung:

Bit 7	Bit 6	Bit 5	Bit 4	Informationsgruppe
0	0	0	1	Position
0	0	1	0	Stadtteil
0	1	0	0	Behörde/öffentl. G.
0	1	1	1	Gesundheitswesen
1	0	0	0	Freizeit
1	0	1	1	Sonstige
1	1	0	1	Gebäudeinfo innen

### 3.5.4 Verkehr

Dieser Kanal hat die Aufgabe, alle notwendigen Informationen über zur Verfügung stehende Verkehrsmittel zu übertragen. Die Information kann in folgende Gruppen unterteilt werden:

Bit 7	Bit 6	Bit 5	Bit 4	Informationsgruppe
0	0	0	1	Verkehrsmittelart
0	0	1	0	Liniennummer
0	1	0	0	Haltestelle
0	1	1	1	Endstationen
1	0	0	0	Allgemeine Info
1	0	1	1	Frei
1	1	0	1	Frei

#### - Allgemeine Information

In dieser Gruppe kann z.B. eine allfällige Störung des normalen Linienbetriebes gesendet werden.

### 3.5.5 Versorgung

Dient zur Übertragung von Hinweisen auf Geschäfte, Restaurants, Kartenschalter usw. Zur Differenzierung werden folgende Gruppen unterschieden:

Bit 7	Bit 6	Bit 5	Bit 4	Informationsgruppe
0	0	0	1	Grundversorgung
0	0	1	0	Freizeit
0	1	0	0	Sanitäre Einrichtungen
0	1	1	1	Sonstige
1	0	0	0	Frei
1	0	1	1	Frei
1	1	0	1	Frei

### 3.5.6 Touristik/Museum/Ausstellung

Enthält Hinweise und Erklärungen zu Museen, historischen Gebäuden, Ausstellungsstücken usw.

### 3.5.7 Datenschnittstellen

Stellt die Kommunikationsschnittstelle zwischen Bankomaten, Fahrkartenautomaten usw. her. Es kann z.B. jedem Bedienvorgang eine Meldung zur Erklärung vorausgehen und eine Meldung als Bestätigung der Ausführung nachfolgen. Hat der Benutzer keinen Erfolg kann er mithilfe der Sonderfunktion 'Alte Meldungen' (siehe Kap. 5.1) den Bedienvorgang nachvollziehen und die entscheidende Fehlbedienung aufspüren.

## **3.6 Benutzerschnittstelle**

Dem Benutzer stehen zwei Bedienelemente zur Beeinflussung des Systems zur Verfügung: Dies sind ein Drehschalter und ein Taster.

### **3.6.1 Der Drehschalter**

Der Drehschalter bietet 16 verschiedene Stellungen. Jeder dieser Stellungen kann ein 'Programm' zugeordnet werden. Zur Zeit werden 14 verschiedene Programme angeboten. Bei Einstellen eines neuen Kanals wird die Funktion ausgesprochen.

Der Drehschalter wird auch zum Auslösen des Programms 'Alte Meldungen' verwendet. Es muß dazu innerhalb einer Sekunde die Schalterstellung verändert und wiederhergestellt werden. Es ertönt dann die Meldung 'Alte Meldungen', und in weiterer Folge werden die gespeicherten Meldungen der Reihe nach gesprochen, wobei nur die Kurzinformation, also nur die wichtigsten Informationsgruppen einer Meldung, gesprochen wird. Will man die gesamte Meldung hören, so kann dies wie im normalen Betrieb durch zweimaliges kurzes Drücken des Tasters veranlaßt werden.

Wurden alle Meldungen gesprochen, so erfolgt die Mitteilung, daß keine weiteren alten Meldungen mehr vorhanden sind und es wird zum normalen Betriebsmodus zurückgeschaltet.

### **3.6.2 Der Taster**

Die Auswertung des Tasters ist so festgelegt, daß sie dem Benutzer zwei verschiedene Arten der Betätigung ermöglicht. Diese werden des weiteren Wahl\_A bzw. Wahl\_B genannt.

Wahl\_A: Der Taster wird einmal lang gedrückt.

Wahl\_B: Der Taster wird zweimal kurz hintereinander gedrückt, wobei er jeweils nicht länger als eine halbe Sekunde gedrückt bleibt.

Der Taster hat im Programm 'Parameter einstellen' andere Funktionen als in den übrigen Programmen.



Funktion Kanal 0-12:

Wahl_A	Wahl_B
überspringe Meldung	wiederhole Meldung

Wurde im Menü 'Parameter einstellen' die Option Kurzinformation(Wirkung siehe Parameter einstellen) aktiviert, so wird beim Wiederholen(Wahl\_B) einer normalerweise davon beeinflussten Meldung die gesamte Meldung ausgesprochen, also die Option für die Dauer der Wiederholung aufgehoben.

Funktion Kanal 13 'Parameter einstellen':

Wahl_A	Wahl_B
Überspringen	Auswählen

### 3.7 Programmauswahl

Dem Benutzer werden zur Selektion der gewünschten Information insgesamt 13 verschiedene Programme und ein Menü zur Parametereinstellung angeboten.

Drehschalterstellung Programm

0	Sicherheit
1	Ampel
2	Orientierung
3	Verkehr
4	Versorgung
5	Touristik/Museum/Ausstellung
6	Information/Datenschnittstelle
7	Einkauf
8	Freizeit
9	Fortbewegung in Gebäuden
10	Orientieren in neuer Umgebung
11	Sanitäre Einrichtungen
12	Hilfe
13	Parameter einstellen

#### 3.7.1 Informationskanäle 0-6

Es werden nur die Meldungen des gewählten Kanals gesprochen. Der Benutzer kann entscheiden, ob er nur die Kurzinformation (siehe Kap. 5.1) hören will oder die gesamte Meldung.

#### 3.7.2 Einkauf

Kombination der Kanäle 'Ampel', 'Orientierung', 'Verkehr' und 'Versorgung'. Es werden dabei bei den Kanälen 'Orientierung' und 'Verkehr' nur Kurzinformationen ausgegeben. Beim Versorgungskanal werden nur jene Informationsgruppen ausgegeben, die zur Gruppe der Grundversorger gehören.

### **3.7.3 Freizeit**

Ist gedacht zur Verwendung bei Freizeitaktivitäten.

In der Grundeinstellung werden die Kanäle Verkehr (kurz), Orientierung (Informationsgruppe Freizeit) und Versorgung (Informationsgruppe Freizeit) ausgegeben.

### **3.7.4 Fortbewegung in Gebäuden**

Es werden nur Meldungen gesprochen, die zur Informationsgruppe 'Gebäudeinformation innen' gehören. Da man davon ausgehen kann, daß bei jedem Eingang zu einem öffentlich zugänglichen Gebäude ein Sender angebracht ist, und damit immer nur aktuelle Informationen zur Orientierung im Gebäude gesprochen werden, wurde auf eine programmtechnische Realisierung dieses Zustands verzichtet.

### **3.7.5 Orientieren in neuer Umgebung**

Es werden die Kurzinformationen der Kanäle 'Ampel', 'Verkehr' und 'Orientierung' ausgegeben. Der Benutzer hat die Möglichkeit durch Tastendruck die gesamte Meldung zu hören.

### **3.7.6 Sanitäre Einrichtungen**

Der Benutzer erhält in diesem Kanal Auskunft über die nächstgelegene Sanitäre Einrichtung. Diese Information wird im Versorgungskanal gesendet.

### **3.7.7 Hilfe**

Es wird der Weg zum nächsten Arzt (Spital) bzw. zur nächsten Polizeistation durchgesagt. Diese Informationen werden im Sicherheitskanal gesendet und sind mit eigenen Kennungen (Erste Hilfe, Polizei) versehen.

### 3.7.8 Parameter einstellen

Das Konzept sieht die Bedienung nur mit dem Taster vor. Es basiert darauf, daß dem Benutzer die jeweiligen Auswahlmöglichkeiten der Menüebene, in der er sich gerade befindet, mitgeteilt werden, und er eine dieser Möglichkeiten auswählt, indem er zum Zeitpunkt, zu dem diese gesprochen wird, oder in der darauffolgenden Pause, den Taster zweimal kurz drückt (Wahl\_B). Wahl\_A (einmal drücken) dient zum Überspringen eines Menüpunktes. Wird keine Taste gedrückt wird nach der Pause automatisch zum nächsten Menüpunkt weitergegangen.

Der Vorteil, den diese Art bietet, liegt darin, daß nicht jedesmal das gesamte Menü durchlaufen werden muß, was besonders zu Geltung kommt, wenn die Einstellmöglichkeiten umfangreich sind bzw. werden sollen/könnten.

Als Nachteil steht dem gegenüber, daß bei zu vielen Menüebenen leicht der Überblick verloren geht.

Es werden vier Parameter zur Anpassung angeboten:

a) Wiederholrate der Meldungen(sehr oft, oft, selten)

Es werden hier zur Zeit drei verschiedene Stufen angeboten. Sie bestimmen die Zeit die bis zur Wiederholung der aktuellen Meldung vergehen soll. Eine Erweiterung auf mehrere Stufen ist jederzeit möglich, scheint jedoch aus Übersichtlichkeitsgründen nicht sinnvoll.

b) Kurzinformation(Ein/Aus)

Ist diese Option aktiviert, so werden nur die wichtigen Informationsgruppen(z.B. nur die Position) ausgesprochen.

Welche Informationsgruppen als Kurzinformation gelten, wird vom Programmierer festgelegt(siehe Kap.8).

c) Dauerton(Ein/Aus)

Der Dauerton hat die Aufgabe dem Benutzer die Möglichkeit zu geben, den Sender, dessen Information gerade gesprochen wird, zu identifizieren. Wird nämlich von diesem Sender gerade etwas empfangen, was immer der Fall ist, wenn der Sender angepeilt wird, und gleichzeitig gesprochen, so ertönt im Hintergrund der Dauerton. Durch die stereophone Wiedergabe kann der Benutzer nun die genaue Position des Senders bestimmen und sich somit auch richtungsmäßig orientieren.

d) Kreiswanderungsüberprüfung(Ein/Aus)

Eine Kreiswanderungsüberprüfung wird ausgelöst, wenn festgestellt wird, daß eine Meldung bereits früher einmal empfangen wurde. Die letzten vier empfangenen Meldungen werden dabei ausgeklammert, um Fehlauflösungen im Streubereich mehrerer

Sender(z.B. an einer Kreuzung) zu vermeiden.

Beim Feststellen einer Kreiswanderung wird dem Benutzer folgende Meldung vorgeschrieben:

"ACHTUNG SIE SIND MÖGLICHERWEISE IM KREIS GEGANGEN. WÄHLEN SIE DEN ORIENTIERUNGSKANAL UND BESTIMMEN SIE IHRE POSITION"

Als Kriterium ob eine Kreiswanderung vorliegt, werden die Zeitspannen, die zwischen den Meldungen liegen, und wie oft die Meldung bereits empfangen wurde, verwendet.

Fallunterscheidung:

Meldung wird das Zweite mal empfangen:

Zeitdifferenz < 5min --> Kreiswanderung

Zeitdifferenz > 5min --> keine Kreiswanderung

Meldung wird das Dritte mal empfangen:

Zeitabstände ca. gleich lang --> Kreiswanderung

Zeitabstände unterschiedlich --> keine Kreiswanderung

Meldung mehr als dreimal empfangen --> Kreiswanderung

#### Beschreibung der Menüebenen:

##### a) Wiederholrate der Meldungen

Es stehen folgende Auswahlmöglichkeiten zur Verfügung:

- sehr oft (alle 20sek)
- oft (alle 40sek)
- selten (alle 80sek)

Die Auswahl erfolgt durch Tastendruck (Wahl\_B), während die gewünschte Wiederholrate gesprochen wird.

##### b) Kurzinformation EIN/AUS

Es wird dem Benutzer der aktuelle Zustand mitgeteilt, und dieser hat die Möglichkeit, diesen durch Tastendruck (Wahl\_B) zu ändern.

##### c) Dauerton EIN/AUS

siehe b)

##### d) Kreiswanderungsüberprüfung EIN/AUS

siehe b)

##### e) Aktuelle Parametereinstellung

Dem Benutzer wird die neue Systemeinstellung mitgeteilt.

## 4. Software

### 4.1 Programmstruktur

Das Programm kann in mehrere Teile gegliedert werden, die jeweils eine Teilaufgabe erledigen.

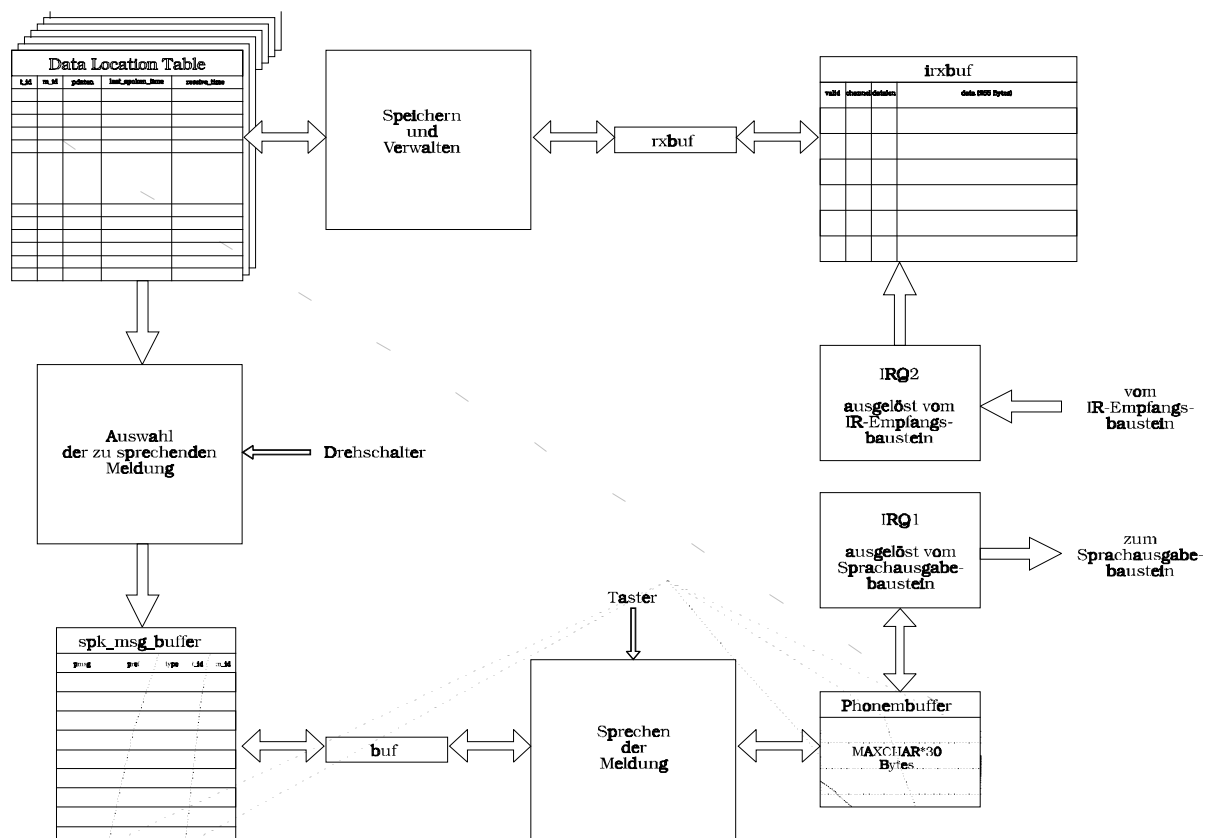
Es werden folgende Aufgaben unterschieden:

- a) Empfangen
- b) Speichern und Verwalten
- c) Auswahl der zu sprechenden Meldungen
- c) Sprechen
- e) Auswertung der Bedienelemente

Einen entscheidenden Einfluß auf den Programmierstil und die Unterteilung in Teilaufgaben nahm die Tatsache, daß ein ständiger Programmdurchlauf gewährleistet sein muß. Es ist dies notwendig, weil ständig Daten empfangen werden und nicht alle Funktionen durch Interrupts implementiert werden können.

Diese Forderung führte zur Einführung mehrerer Buffer, Tabellen und Systemvariablen, die den Daten- und Informationsaustausch zwischen den Modulen bewerkstelligen.

Das folgende Diagramm soll das Zusammenspiel der einzelnen Module veranschaulichen:



## 4.2 Speicherverwaltung

Einer der wesentlichsten Teile des Programmes ist die Zwischenspeicherung und Verwaltung der empfangenen Daten. Es ist dabei zu berücksichtigen, daß nur 32kB RAM zur Verfügung stehen, und daher der Speicher gut auszunützen ist. Weiters soll die Verwaltung so erfolgen, daß ein Rückverfolgung des Weges anhand der Meldungskennung(T-ID,M-ID) ermöglicht wird, um eventuelle Kreiswanderungen des Benutzers erkennen zu können.

Um das am besten geeignete Verfahren zu finden, wurden die Vorteile der einzelnen Verfahren (Tabellen, verkettete Listen, Baumstrukturen) den Anforderungen des Programms und den Hardwareeinschränkungen gegenübergestellt.

Aufgrund der Anforderungen des Programms standen verkettete Listen(einfach bzw doppelt verkettete) und Tabellen zur Auswahl.

Baumstrukturen konnten von der näheren Betrachtung ausgeschlossen werden, da ihre Vorteile hauptsächlich bei der Suche nach bestimmten Einträgen hervortreten, und dies in dieser Anwendung kaum vorkommt.

Es zeigte sich, daß sich Vor- und Nachteile bezüglich Programmierung in etwa die Waage hielten. Damit war der benötigte Speicherplatz das entscheidende Auswahlkriterium, womit der einfach verketteten Liste gegenüber der doppelt verketteten der Vorzug zu geben ist.

Um den wahren Bedarf an Speicherplatz der Verfahren feststellen zu können, war eine genauere Betrachtung der interne Speicherverwaltung der verwendeten Programmiersprache C nötig.

Dies zeigte, daß der Speicher in Blöcke zerlegt ist, die miteinander einfach verkettet sind. Zur Verwaltung besitzt jeder Block einen Header, der den Zustand des Blocks (frei oder belegt) und einen Zeiger auf den nächsten Block beinhaltet, d.h. der Header hat im Fall der verwendeten CPU 6303 eine Größe von 3Byte.

Das bedeutet, daß auch bei jeder Speicherallokierung ein 3Byte-Header, unabhängig von der Größe des allokierten Speichers, miterzeugt wird, da C eine Fragmentierung des Speichers bis hinzu 1Byte-Blöcken erlaubt.

Bei der einfach verketteten Liste wird nun bei Aufnahme einer neuen Meldung jedesmal ein Speicherblock mit 14Byte Größe angefordert. Dabei werden natürlich jedesmal die 3Byte für den Header mitbelegt.

Bei der Tabelle wird abhängig von der Größe der Tabelle (Anzahl der Einträge  $n$ ) ein Speicherblock mit der Größe  $n \cdot 12$ Byte angefordert. Nimmt man 32 Einträge als Basiswert,

so ergibt das einen Block mit 384Byte Größe + 3Byte Header (nur einmal!).

Damit zeigt sich, daß bei Verwendung der von C zur Verfügung gestellten Speicherverwaltungsroutinen, einer Tabelle eindeutig der Vorzug zu geben ist. Die Verwendung einer einfach verketteten Liste erscheint nur dann sinnvoll, wenn man einen größeren Speicherblock anfordert und diesen selbst mit Hilfe einer geeigneten Zeigerarithmetik verwaltet und damit das Problem mit den Headern umgeht. Da der damit verbundene Mehraufwand jedoch das Programm verlängern und schwerer lesbar machen würde, wird zur Verwaltung eine Tabelle verwendet, die wie ein Ringpuffer behandelt wird.

Data\_Location\_Table :

T_ ID	M_ ID	Data- pointer	last_time_spoken	receive_time

Zur Tabellenverwaltung werden zwei Zeiger benötigt:

**Last\_Valid** zeigt auf das letzte gültige Element in der Liste. Wird um 1 erhöht, wenn der Eintrag, auf den gerade gezeigt wird, seine Gültigkeit verliert.

**Level** entspricht der Anzahl der gültigen Tabelleneinträge minus 1 (0 = 1Eintrag).

Es wird für jeden Informationskanal eine Data\_Location\_Table angelegt. Es werden jedoch für die Kanäle 'Sicherheit' und 'Ampel' nur Tabellen mit der Größe 1 angelegt. Für die Kanäle 'Touristik' und 'Datenschnittstelle' werden vorerst ebenfalls nur Tabellen dieser Größe angelegt. Erst bei Verwendung dieser Kanäle werden normal große Tabellen angelegt. Es wird



dadurch unnötiger Speicherplatzverbrauch vermieden. Die Daten werden im gemeinsamen verbleibenden Speicher abgelegt.

Meldungen, die aus mehreren Blöcken bestehen, werden erst dann in die Liste aufgenommen und im Speicher abgelegt, wenn alle Blöcke empfangen worden sind.

Um unnötige Speicherbelegung zu vermeiden, wird überprüft, ob die abzuspeichernde Meldung bereits einmal empfangen wurde (Suchschlüssel sind T\_ID und M\_ID). Ist dies der Fall, so wird lediglich der Zeiger auf die Daten kopiert (innerhalb der DLT).

Fallunterscheidungen für das Einfügen einer empfangenen Meldung:

1. Die Meldung ist nicht unter den ersten 4 Meldungen enthalten
  - > Meldung wird in Tabelle aufgenommen -> es wird überprüft, ob die Meldung bereits früher einmal empfangen wurde
    - nein: Speicher wird angefordert und Daten abgelegt. Tritt ein Speicheranforderungsfehler auf, so werden alte Meldung in der Tabelle gelöscht und der von ihnen belegte Speicher freigegeben.
    - ja: Es wird nur der Zeiger auf die Daten kopiert . Es wird außerdem überprüft, ob eine Kreiswanderung vorliegt.
  
2. Die Meldung ist unter den ersten 4 Meldungen enthalten -> Die Meldung wird an die Tabellenspitze verschoben. Damit wird ein Überlaufen der Tabelle und das damit verbundene "Verschütten" älterer Meldungen verhindert.

### Abschätzen der benötigten Tabellengröße

Zur Abschätzung werden folgende Annahmen getroffen:

1. Die durchschnittliche Länge einer Meldung betrage 128Byte.
2. Pro Transmitter werden durchschnittlich 4 Meldungen empfangen.
3. Von den vorhandenen 32kB RAM bleiben ca. 24kB für die Abspeicherung der Meldungen übrig.

Diese Annahmen ergeben eine mittlere benötigte Anzahl von 48 Einträgen pro Tabelle.

Um unnötige Speicherbelegung weitgehend zu minimieren, werden die Tabellen vorerst mit

einer Größe von 32 Einträgen allokiert und bei Bedarf erweitert. Da die Erweiterung einen nicht unbeträchtlichen Aufwand an Programmlaufzeit erfordert, wird immer in 16er-Blöcken erweitert, wobei eine Beschränkung auf maximal 64 Einträge pro Tabelle erfolgt.

### **4.3 Empfangen**

Da es sich um ein serielles Übertragungsprotokoll handelt und die Meldungsblöcke daher byteweise empfangen werden, wird diese Aufgabe von einer Interruptroutine (IRQ2\_interrupt) übernommen. Der Empfangsbaustein löst den Interrupt aus, wenn wieder ein Byte empfangen wurde. Die Interruptroutine filtert die zum Protokoll gehörenden Bytes aus, und legt den empfangenen Block im Empfangsbuffer ab. Es gibt für jeden Kanal einen Buffer von der maximalen Größe eines Blocks. Dies ermöglicht den Empfang aller Meldungen auch bei relativ kurzem Hinsehen zu einem Sender.

Das Vorhandensein einer neuen gültigen (d.h. vollständig empfangenen) Meldung in einem Buffer wird durch ein Flag signalisiert. Das Hauptprogramm überprüft die Empfangsbuffer und übergibt in der Folge den nächsten gültigen Block an den Buffer des Verwaltungsprogrammes, und gibt den zugehörigen Kanal im Empfangsbuffer wieder frei. Dieser, auf den ersten Blick nicht sinnvoll erscheinende Buffer für das Verwaltungsprogramm, erlaubt das rasche Empfangen von Meldungen, die aus mehreren Blöcken bestehen, weil in der Zeit, in der eine Block bearbeitet wird, bereits der nächste empfangen werden kann.

### **4.4 Speichern und Verwalten**

Der im Buffer des Verwaltungsprogramms befindliche Block wird zuallererst auf seine Gültigkeit geprüft, indem die XOR-Verknüpfung der Daten und die negierte Summe aus den Daten und dem empfangenen XOR-Byte gebildet wird. Diese werden mit den übertragenen Prüfbytes verglichen.

Bei Erfolg wird anschließend zwischen Meldungen, die aus einem oder mehreren Blöcken bestehen, unterschieden.

Meldungen, die aus mehreren Blöcken bestehen, werden vorerst in einem Zwischenspeicher abgelegt, und erst dann in die Speicherverwaltungstabelle aufgenommen, wenn sie vollständig empfangen wurden. Die Reihenfolge, in der die Blöcke dabei empfangen werden, ist beliebig. Meldungen, die nur aus einem Block bestehen, werden natürlich sofort aufgenommen.

Das Einfügen in die Speicherverwaltungstabelle erfolgt nach den in Kap.4.2 erläuterten Regeln.

Es werden dazu folgende Funktionen benötigt:

- neue Meldung eintragen (add\_entry)
- Daten speichern (save)
- Eintrag verschieben (move\_entry)
- Tabellengröße verändern (resize\_dlt)
- Speicher freigeben (release\_mem bzw. free\_mem)

Die Funktion 'Tabellengröße verändern' wird aufgerufen, wenn beim Eintragen einer neuen Meldung festgestellt wird, daß die Tabelle bereits voll ist. Ist jedoch bereits die maximale Tabellengröße erreicht, werden Meldungen, die älter als 30min sind, gelöscht. Erfüllt keiner der Einträge dieses Kriterium, wird der älteste Eintrag gelöscht. Dieser Vorgang wird auch ausgelöst, wenn zuwenig Speicherplatz für die Daten einer neuen Meldung vorhanden ist.

## 4.5 Auswahl der zu sprechenden Meldungen

Die Auswahl der nächsten zu sprechenden Meldung wird hauptsächlich von der Bedienung des Drehschalters und des Tasters bestimmt.

Beim Drehschalter werden neben der Stellung(0-15) noch drei Zustände unterschieden:

- Schalterstellung unverändert (SWT\_NOT\_CHANGED)
- Schalterstellung verändert (SWT\_CHANGED)
- Alte Meldungen erwünscht (OLD\_MSG\_REQ)

Beim Taster werden 4 verschiedene Zustände unterschieden:

- nicht gedrückt (NO\_KEY)
- Fehlbedienung (KEY\_ERROR)
- einmal gedrückt (WAHL\_A)
- zweimal kurz gedrückt (WAHL\_B)

In Abhängigkeit dieser Zustände erfolgt die Auswahl der nächsten zu sprechenden Meldung bzw. der Reaktion auf diesen Zustand.

In diesem Modul werden auch die kombinierten Programme und das Menü zur Parametereinstellung realisiert.

## 4.6 Sprechen

Das Sprachausgabemodul ist so realisiert, daß es unabhängig von den anderen Modulen arbeitet. Es teilt den übrigen Modulen seinen Zustand über globale Variablen mit. Die Schnittstelle stellt der Sprachausgabebuffer (spk\_msg\_buf) dar. Ist die Ausgabe einer Meldung beendet, wird aus dem Buffer die nächste zu sprechende Meldung entnommen und abgearbeitet. Zum Füllen des Buffers wird eine eigene Funktion angeboten, die, zur besseren Erkennbarkeit der Funktion, beim Lesen des Programms nicht 'write' sondern 'spk' genannt wurde.

Es werden zwei verschiedene Typen von Meldungen unterschieden:

- Programmierungen (PROGRAM\_MSG)
- Informationen (INFORMATION)

Der prinzipielle Ablauf erfolgt so, daß die Meldung abschnittsweise (im Normalfall wortweise) in Phonemdaten übersetzt wird, diese in einem Buffer abgelegt werden und dann durch eine Interruptroutine an den Sprachausgabebaustein ausgegeben werden. Der Sprachausgabebaustein löst dabei jedesmal, wenn ein Phonem abgearbeitet ist, den Interrupt aus. Dieser sperrt sich selbst, sobald alle Phoneme im Buffer ausgegeben wurden, und teilt dies über eine globale Variable mit, wodurch der beschriebene Vorgang von neuem begonnen wird, wenn das Ende der Meldung noch nicht erreicht ist.

Die Sprachsynthesoftware zum Übersetzen in Phonemdaten wurde als Diplomarbeit von Walter Authried (/AUT89/) entwickelt.

Zur Beeinflussung der Sprachausgabe werden noch zwei weitere Funktionen zu Verfügung gestellt:

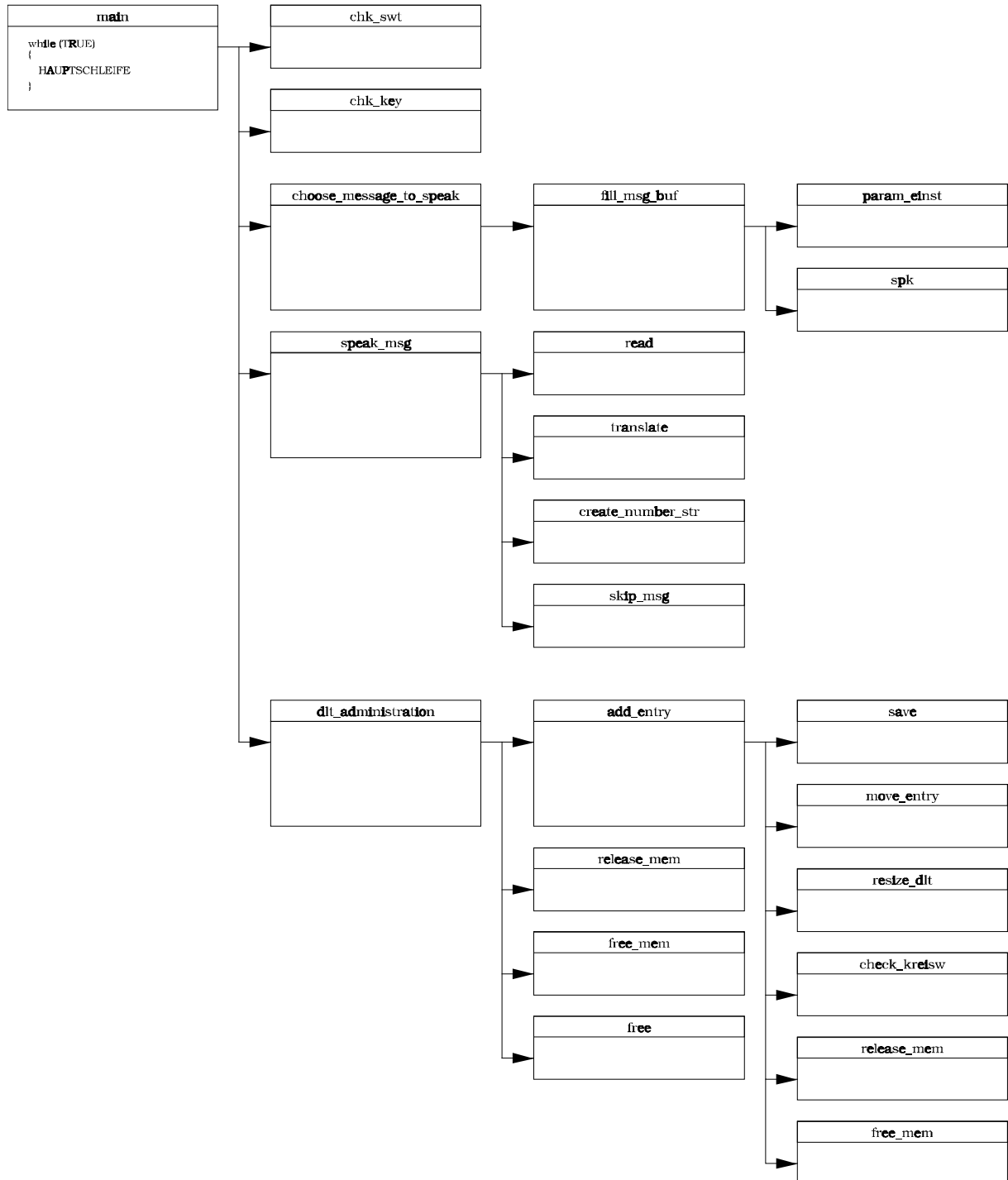
- skip\_msg ... beendet die aktuelle Ausgabe einer Meldung
- clear\_msg\_buf ...leert zusätzlich noch den Sprachausgabebuffer

## 4.7 Auswerten der Bedienelemente

Dem Benutzer stehen, wie bereits in Kap.3.6 genauer erläutert, ein Taster und ein Drehschalter zur Bedienung des Geräts zur Verfügung. Die Bedeutung, die diese Elemente erhalten, hängt von der Art ihrer Betätigung ab. Die Auswertung hängt dabei im wesentlichen von der Zeit ab. Da es sich dabei um kurze Zeitintervalle handelt, war es notwendig eine Systemzeit mit einer Auflösung von einer Zehntelsekunde zu erzeugen. Da diese Systemzeit auch zur Verwaltung und zur Meldungswiederholung verwendet wird und vor allem das Verwaltungsprogramm auf großen 'Zeitkonstanten' (30 min) aufbaut, wurde als Variablentyp 'unsigned long' gewählt. Dieser gewährleistet, daß kein Überlauf auftreten kann und damit keine unerwünschten Aktionen ausgelöst werden,

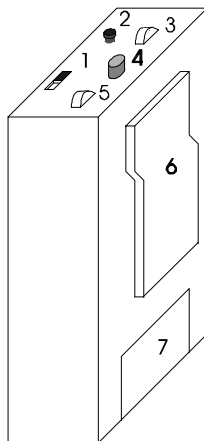
Es wäre auch möglich, eine Betriebsstundenansage zu implementieren, wenn dies gewünscht wird.

## 4.8 Funktionsaufrufe



## 5. Bedienungsanleitung

Das Infrarot-Informations- und Orientierungssystem 'I.R.I.S' dient zur Mobilitätsverbesserung von Blinden und Sehbehinderten im Nahverkehr. Es ermöglicht den Zugang zu Informationen die ansonst nicht ohne fremde Hilfe in Erfahrung gebracht werden können. Das System besteht aus fest installierten Sendern und dem Empfänger den der Behinderte mit sich trägt. Im Empfänger, der wie ein 'Walkman' am Gürtel befestigt werden kann, sind die Bedienelemente untergebracht. Es stehen ein Taster und ein Drehschalter zur Beeinflussung des Systems zur Verfügung. Diese Beschränkung auf sehr wenige Bedienelemente ermöglicht eine einfache Handhabung des Geräts und verkürzt die 'Einlernphase' wesentlich.



- 1...Ein/Aus Schalter
- 2...Taster
- 3...Sprachlautstärke
- 4...Drehschalter
- 5...Dauertonlautstärke
- 6...Gürtelhalter (Clip)
- 7...Batteriefach (9V)

Es gibt zwei Potentiometer zur Regelung der Lautstärke. Das an der linken Seite angebrachte Potentiometer (5) dient zur Lautstärkeeinstellung des Dauertons, während mit dem rechts untergebrachten (3) die Lautstärke der synthetischen Sprache verändert werden kann. Die Funktionen des Tasters und des Drehschalters werden später besprochen.

Ehe auf das Programmangebot und die Bedienelemente Taster und Drehschalter genauer eingegangen wird, sollen die Sonderfunktionen

- Kurzinformation
- Dauerton
- 'Alte Meldungen'
- Alarmmeldung
- Kreiswanderungsüberprüfung

näher erläutert werden.

## 5.1 Sonderfunktionen

### Kurzinformation

Da viele Meldungen Informationen enthalten die entweder oft gesendet werden und daher hinlänglich bekannt sind und deshalb als überflüssig erachtet werden oder lediglich Zusatz- bzw. Detailinformationen darstellen, wurde die Möglichkeit geschaffen nur die wesentlichen Informationen sprechen zu lassen.

Es soll dies im folgenden anhand einer typischen Meldung im Kanal 'Verkehr' gezeigt werden:

Kurzinformation AUS: "RECHTS UBAHN LINIE U1 HALTESTELLE TAUBSTUMMENGASSE NACH REUMANNPLATZ UND KAGRAN"

Kurzinformation EIN: "RECHTS UBAHN LINIE U1 HALTESTELLE TAUBSTUMMENGASSE"

### Dauerton

Um zu wissen in welche Richtung man gehen muß um an bestimmte Orte (Räume, Gebäude, Geschäfte,...) zu gelangen ist es unbedingt notwendig einen Anhaltspunkt zu haben an dem man sich orientieren kann. Da das einzig zur Verfügung stehende Objekt der Sender ist wird dieser mithilfe des Dauertons und der stereophonen Sprachausgabe zum Referenzpunkt gemacht. Das Prinzip ist sehr einfach und basiert darauf, daß im Hintergrund ein Ton konstanter Höhe (der Dauerton) zu hören ist, wenn von einem Sender Informationen empfangen werden und im gleichen Augenblick eine von diesem Sender stammende Meldung gesprochen wird. Damit kann nun der Sender von dem diese Meldung stammt identifiziert werden und sein ungefährender Standort bestimmt werden. Die genaue Standortbestimmung erfolgt mithilfe der stereophonen Wiedergabe die nur dann eine Meldung in beiden Ohrhörern mit gleicher Lautstärke wiedergibt wenn der Sender gerade angesehen wird. D.h. wenn man den Kopf beispielsweise nach links dreht wird die Meldung im rechten Ohr lauter zu hören sein als im linken und umgekehrt. Damit ist die genaue Peilung des Senders möglich und man ist nun in der Lage sich an der gesendeten Meldung zu orientieren.



### **'Alte Meldungen'**

Ein besonderes Merkmal des Empfängers ist die Tatsache, daß er die empfangenen Meldungen abspeichert und zur späteren Auswertung aufhebt. Wie weit die gespeicherten Meldungen zeitmäßig zurückreichen hängt davon ab wieviele Sender man passiert hat. Es werden bis zu 64 Meldungen pro Kanal abgespeichert, was einem Zeitraum von schätzungsweise 60-90 min entspricht wenn man von einem dichten Sendernetz ausgeht.

Will man nun diese 'Alten Meldungen' abhören so dreht man den Drehschalter ein Schalterstellung vor (oder zurück) und stellt anschließend (innerhalb einer Sekunde) die Ausgangsstellung wieder her. Es ertönt dann die Meldung 'ALTE MELDUNGEN', und es werden die gespeicherten Meldung der Reihe nach gesprochen, wobei bei der 'jüngsten' begonnen wird. Die 'jüngste' ist dabei nicht die zuletzt empfangene Meldung (das ist die aktuelle Meldung) sonder die davor empfangene, d.h. die aktuelle Meldung wird nicht gesprochen. Es wird dabei nur die Kurzinformation gesprochen wobei jedoch durch zweimaliges kurzes Drücken des Tasters eine Wiederholung der gerade gesprochenen Meldung erfolgt und diese dann in voller Länge wiedergegeben wird.

Durch einmaliges Drücken des Taster wird die Ausgabe vorzeitig abgebrochen und zum normalen Betrieb zurückgekehrt.

### **Alarmmeldung**

Alarmmeldungen sind Meldungen die in Notsituationen (z.B. im Brandfall) gesendet werden. Sie werden unabhängig vom gerade ausgewählten Kanal gesprochen und zwar solange bis man das Sendefeld verlassen hat. Ab diesem Zeitpunkt ist die Alarmmeldung nur mehr im Sicherheitskanal abrufbar. Alarmmeldungen können während der Ausgabe, wie jede andere Meldung auch, durch einmaliges Drücken des Tasters abgebrochen und durch zweimaliges Drücken wiederholt werden.

### **Kreiswanderungsüberprüfung**

Die Kreiswanderungsüberprüfung ist als eine unterstützende Funktion zur Orientierung in neuer Umgebung, insbesondere in Gebäuden, gedacht und soll darauf aufmerksam machen, daß man einen Ort schon mehrmals passiert hat oder erst vor kurzem an diesem Ort war. Wird von Programm eine vermutliche Kreiswanderung festgestellt erfolgt die Durchsage folgender Meldung: "ACHTUNG SIE SIND MÖGLICHERWEISE IM KREIS GEGANGEN. WÄHLEN SIE DEN ORIENTIERUNGSKANAL UND BESTIMMEN SIE IHRE POSITION".

Diese Sonderfunktion kann im Menü 'Parameter einstellen' aus- bzw. eingeschaltet werden.

## 5.2 Die Bedienelemente

Wie bereits eingangs dieses Kapitels erwähnt stehen ein Taster und ein Drehschalter zu Bedienung zur Verfügung. In Kap. 3.6 wurden die verschiedenen Möglichkeiten zur Benutzung erläutert, daher soll hier gezeigt werden was man tun muß um bestimmte Programmkaktivitäten auszulösen.

### a) Taster

Es muß zwischen dem normalen Betrieb und dem Programm 'Parameter einstellen' unterschieden werden:

- Normaler Betrieb
  - Ausgabe einer Meldung abbrechen ●
  - Meldung wiederholen ●●
- 'Parameter einstellen'
  - Auswählen einer Einstellung bzw. ändern der aktuellen Einstellung ●●
  - Weiter zum nächsten Menüpunkt ●

● ... bedeutet einmaliges Drücken des Taster

●●... bedeutet zweimaliges kurzes Drücken des Tasters

Ausgabe einer Meldung abbrechen ●

Diese Interaktion führt man durch wenn man die Meldung die gerade gesprochen wird bereits kennt, die Meldung nicht von Interesse ist oder man gerade 'Alte Meldungen' abhört und die gesuchte schon gehört hat und daher wieder zum normalen Betriebsmodus zurückkehren möchte.

Meldung wiederholen ●●

Hat man die Sonderfunktion 'Kurzinformation' aktiviert und wird eine Meldung gesprochen die von Interesse ist und von der man auch die Zusatz- bzw. Detailinformationen wissen will so erreicht man das durch diese Bedienung des Tasters, weil beim Wiederholen einer Meldung immer die gesamte Meldung gesprochen wird.

**b) Drehschalter**

Mit dem Drehschalter können 16 Drehschalterstellungen angewählt werden, wovon zur Zeit 14 belegt sind (Zuordnung siehe Kap. 3.7).

Es gibt zwei mögliche Gründe zur Veränderung der Drehschalterstellung:

- Auswählen eines neuen Programms

Bei Änderung der Drehschalterstellung wird das der neuen Stellung zugeordnete Programm angesagt, d.h. Sie drehen den Drehschalter einfach solange weiter bis das gewünschte Programm erreicht ist.

- Aktivieren des Programms 'Alte Meldungen'

Drehen Sie den Drehschalter eine Rasterstellung nach vor und innerhalb einer Sekunde wieder zurück (oder umgekehrt).

Beim Auswählen eines neuen Programms ist zu Bedenken, daß die Ausgabe einer Meldung dadurch abgebrochen wird, d.h. wenn Sie die gerade gesprochene Meldung zu Ende hören wollen sollten Sie mit der Änderung der Drehschalterstellung bis zum Ende der Meldung zuwarten.

## 6. Programmbeschreibung

### 6.1 Allgemeines

In Kapitel 4.1 wurden die Anforderungen an das Programm und die daraus resultierende Programmstruktur bereits näher beschrieben. Dieses Kapitel befaßt sich mit der programmtechnischen Lösung der Teilaufgaben, ihrer Zusammenfassung in logischen Einheiten sowie dem Datenaustausch zwischen den Programmen.

Das Programm wurde in fünf Programmdateien und mehrere dazugehörige Bibliotheken aufgeteilt:

- IRIS.C
- RXD.C
- SPEAK.C
- USER.C
- TIME.C

Ehe jedoch zur Detailbeschreibung der einzelnen Programme übergegangen wird werden die wichtigsten Systemvariablen - sie bilden den Systemzustand ab - und die Buffer erläutert, die einen großen Teil des Datenaustauschs bewerkstelligen.

Da diese Systemvariablen meist Zustände und keine wirklichen Zahlenwerte darstellen werden aussagekräftige 'Zustandsnamen' zugewiesen, welche in der Bibliothek CONST.H definiert sind.

## 6.2 Systemvariablen

```
param_type param;          typedef struct
                            {
                                unsigned int rep_time;
                                unsigned char k_info,
                                unsigned char dauerton;
                                unsigned char kreiswanderung;
                            } param_type;
```

*param* enthält die vom Benutzer in Menü 'Parameter einstellen' gewählte Systemeinstellung.

param.rep_time	Wiederholrate der Meldungen in Zehntelsekunden
param.k_info	Kurzinformation EIN/AUS
param.dauerton	Dauerton EIN/AUS
param.kreiswanderung	Kreiswanderungsüberprüfung EIN/AUS

char *key*;

*key* enthält den Zustand des Tasters. Es werden vier verschiedenen Zustände unterschieden:

NO_KEY	Taster wurde nicht betätigt
KEY_ERROR	Taster wurde ungültig betätigt
WAHL_A	Taster wurde einmal gedrückt
WAHL_B	Taster wurde zweimal kurz hintereinander gedrückt

char *swt*;

*swt* enthält die Stellung des Drehschalters. Es gibt 16 Schalterstellungen wovon zur Zeit 14 ein Programm zugewiesen wird:

0 SICHERHEIT	7 EINKAUF
1 AMPEL	8 FREIZEIT
2 ORIENTIERUNG	9 FORTB_IN_GEB
3 VERKEHR	10 NEU_ORIENT
4 VERSORGUNG	11 SANITARY
5 TOURISTIK	12 HILFE
6 DATENSCHN	13 PARAM_EINST

char *swt\_status*;

*swt\_status* enthält den Zustand des Drehschalters. Es werden drei verschiedenen Zustände unterschieden:

SWT_NOT_CHANGED	Drehschalterstellung unverändert
SWT_CHANGED	Drehschalterstellung wurde verändert
OLD_MSG_REQ	Drehschalterstellung wurde verändert und wiederhergestellt, d.h. das Programm 'Alte Meldungen' wird gewünscht

unsigned char *dauerton*;

*dauerton* ist TRUE, wenn die Bedingung "Die Meldung die gerade gesprochen wird stammt von dem Sender von dem gerade Informationen empfangen werden." erfüllt ist.

unsigned long int *actual\_time*;

*actual\_time* enthält die aktuelle Systemzeit (sie ist während eines Programmdurchlaufes konstant) in Zehntelsekunden. Die Systemzeit ist jene Zeit, die seit dem Einschalten des Empfängers vergangen ist.

unsigned char *speaker\_busy*;

*speaker\_busy* ist TRUE, wenn die im Phonembuffer abgelegten Daten noch nicht vollständig an den Sprachausgabebaustein ausgegeben sind.

unsigned char *talker\_busy*;

*talker\_busy* ist TRUE, wenn eine Meldung noch nicht vollständig in Phoneme übersetzt ist.

unsigned char *old\_msg*;

*old\_msg* ist TRUE, wenn das Programm 'Alte Meldungen' abgearbeitet wird.

unsigned char *repeat\_msg*;

*repeat\_msg* wird TRUE gesetzt wenn eine Meldung auf Wunsch des Benutzers wiederholt werden soll. Dies hat zur Folge, daß die Meldung auch bei 'Kurzinformation EIN' zur Gänze gesprochen wird.

```

regbytes_type regbytes;          typedef struct
                                  {
                                  char   *fortsetz;
                                  int    zahl;
                                  } regbytes_type;

```

*regbytes* enthält die zur weiteren Übersetzung eines Textes notwendigen Informationen.

*regbytes.fortsetz* zeigt auf jene Stelle im Text, bis zu der (exklusive) die zugehörigen Registerwerte (für den Sprachsynthesebaustein) bereits ermittelt wurden. Wenn das Textende erreicht ist, wird in 'fortsetz' 0 übergeben.

*regbytes.zahl* gibt an, wieviele Phoneme zu je 5 Registerbytes im Phonembuffer abgelegt worden sind.

```
unsigned char act_rxd_tid;
```

```
unsigned char act_rxd_mid;
```

*act\_rxd\_tid* und *act\_rxd\_mid* enthalten die Identifizierung des Senders von dem gerade Meldungen empfangen werden.

### 6.3 Buffer

Die nachfolgend erläuterten Buffer bewerkstelligen größtenteils den Datenaustausch zwischen den Modulen. Ihre Position in der 'Datenflußkette' kann der Abbildung auf Seite 18 entnommen werden.

```

rxbuf_type rxbuf;          typedef struct
                            {
                                char          valid;
                                unsigned char channel;
                                unsigned char datalen;
                                unsigned char data[255];
                            } rxbuf_type;

```

*rxbuf* stellt die Verbindung zwischen der Empfangsroutine (IRQ2\_interrupt) und der Speicherverwaltungsroutine (dlt\_administration) dar.

rxbuf.valid	zeigt an, ob der Buffer eine Meldung enthält.
rxbuf.channel	enthält die Kanalnummer (0-6).
rxbuf.datalen	enthält die Länge der Meldung in Bytes (inkl. Kennungs- und Prüfbytes).
rxbuf.data	enthält den empfangenen Meldungsblock (siehe auch Abb. 3).

```
rxbuf_type irxbuf[7];
```

Die Felder von *irxbuf* haben die gleiche Struktur wie *rxbuf*. Sie stellen den Empfangsbuffer dar, welcher von der Interruptempfangsroutine (IRQ2) gefüllt wird.



```

dlt_zeiger_type dlt_zeiger[7];           typedef struct
                                           {
                                           dlt_type           *pdlt;
                                           unsigned char     last_valid;
                                           char               dlt_level;
                                           unsigned char     dlt_size;
                                           } dlt_zeiger_type;

```

Der Struct *dlt\_zeiger* dient als Verwaltungstabelle für die Data Location Tables (DLT), welche als Tabelle zur Speicherverwaltung dient. Da für jeden Informationskanal eine DLT angelegt wird, werden sieben Verwaltungstabellen benötigt.

dlt_zeiger.pdlt	zeigt auf die zugehörige DLT.
dlt_zeiger.last_valid	zeigt auf das letzte gültige Element in der DLT.
dlt_zeiger.dlt_level	enthält die Anzahl der gültigen Tabelleneinträge minus 1.
dlt_zeiger.dlt_size	enthält die grÖÙe der DLT, d.h. die maximale Anzahl der ablegbaren Meldungen.

Die DLT's werden dynamisch angelegt um sie jederzeit bei Bedarf vergrÖÙern zu können. Sie werden wie folgt vereinbart:

```

pdlt = (dlt_type *) malloc ( sizeof (dlt_type) * n);   n...entspricht der gewünschten
                                                       GrÖÙe der DLT.

```

```

typedef struct
{
  unsigned char     t_id;
  unsigned char     m_id;
  char              *pdaten;
  unsigned long int last_time_spoken;
  unsigned long int receive_time;
} dlt_type;

```

dlt_type.t_id	enthält die Senderidentifizierung.
dlt_type.m_id	enthält die Meldungsidentifizierung.
dlt_type.pdaten	zeigt auf die Daten der Meldung (der benötigte Speicherplatz wird wie bei den DLT's über 'malloc' dynamisch angelegt).

dlt\_type.last\_time\_spoken enthält die Systemzeit bei der die Meldung zuletzt gesprochen wurde.

dlt\_type.receive\_time enthält die Systemzeit bei der die Meldung empfangen wurde.

```

spk_msg_buf_type spk_msg_buf[10];    typedef struct
                                        {
                                            char          *pmsg_to_spk;
                                            char          *pref_tab;
                                            char          type_of_msg;
                                            unsigned char tid;
                                            unsigned char mid;
                                        } spk_msg_buf_type;

```

Der Sprachausgabebuffer *spk\_msg\_buf* stellt die Schnittstelle des Sprachausgabemoduls zu den übrigen Modulen her. Es wird die Funktion 'spk' zum Füllen des Buffers zur Verfügung gestellt.

Das Sprachausgabemodul arbeitet den Buffer der Reihe nach (FIFO) ab.

spk\_msg\_buf.pmsg\_to\_spk zeigt auf die zu sprechende Meldung.

spk\_msg\_buf.pref\_tab zeigt auf die Referenztabelle (wird für die Sonderfunktion 'Kurzinformation' benötigt).

spk\_msg\_buf.type\_of\_msg enthält den Meldungstyp. Es werden zwei Typen unterschieden: PROGRAM\_MSG Programmierungen  
INFORMATION Informationen

spk\_msg\_buf.tid enthält die Senderidentifizierung.

spk\_msg\_buf.mid enthält die Meldungsidentifizierung.

```
spk_msg_buf_type buf;
```

*buf* dient dem Sprachausgabemodul als Arbeitsbuffer.

```
unsigned char phonem_buffer[MAXCHAR*30];
```

Im *phonem\_buffer* werden die Registerwerte für den Sprachsynthesebaustein abgelegt. MAXCHAR legt die maximale Anzahl der Zeichen fest die in einem Durchlauf übersetzt werden können.

## 6.4 IRIS.C

IRIS.C stellt das Hauptprogramm dar und beinhaltet daher die Initialisierung der Systemkomponenten und die Endlosschleife für den immerwährenden Programmdurchlauf.

Des Weiteren wird in IRIS.C die Systemzeit gebildet, der Dauerton ein bzw. ausgeschaltet und der Empfangsbuffer auf neue Meldungen hin überprüft.

Das Flußdiagramm kann in Anhang B eingesehen werden.

## 6.5 RXD.C

Die wichtigsten Routinen dieses Programmmoduls sind das Verwaltungshauptprogramm (*dlt\_adminstration*), die dazugehörigen Speicherfunktionen (*add\_entry*, *move\_entry*, *save*, *free\_mem*), die Interruptempfangsroutine (*IRQ2\_interrupt*), sowie die Kreiswanderungsüberprüfung (*check\_kreisw*).

Flußdiagramme des Verwaltungsprogramms und der Kreiswanderungsüberprüfung finden sich in Anhang B.

## 6.6 SPEAK.C

In SPEAK.C sind alle zur Sprachausgabe benötigten Routinen (*spk\_msg*, *spk*, *read*, *create\_number\_str*), und die dazu benötigte Interruptroutine (*IRQ1\_interrupt*) untergebracht. Weiters enthält dieser Programmteil die Interruptroutine zur Erzeugung des Dauertons (*TOC2\_interrupt*), das Programm für die Parametereinstellung durch den Benutzer (*param\_einst*), sowie die Realisierung der Menüs, in Form der Prozeduren *choose\_message\_to\_speak* und *fill\_msg\_buf*. Flußdiagramme der Prozeduren *spk\_msg* und *choose\_message\_to\_speak* finden sich in Anhang B.

Zur Erzeugung der Programme und der Kurzinformation ist es teilweise notwendig, daß nur bestimmte Informationsgruppen gesprochen werden. Die Ausblendung unerwünschter Informationsgruppen, welche in *spk\_msg* durch &-Verknüpfung mit dem der Informationsgruppe entsprechenden Eintrag in der dem ausgewählten Programm zugehörigen Referenztable (siehe Anhang A; 0xFF kennzeichnet jene Informationsgruppen, die zu sprechen sind) erfolgt.

## **6.7 USER.C**

In diesem Programmteil befinden sich die Funktionen zur Auswertung der Bedienelemente (*chk\_swt, chk\_key*).

Die zugehörigen Flußdiagramme sind in Anhang B zu finden.

## **6.8 TIME.C**

TIME.C enthält die Interruptroutine (*TOI\_interrupt*) die zur Erzeugung der Systemzeit notwendig ist, sowie die dazugehörige Initialisierungsroutine (*clock\_init*).

## 7. Senderprogrammierung

Die Programmierung der Sender ist im Bezug auf die Effektivität der im Empfänger zur Verfügung gestellten Programme von großer Bedeutung, da nur dann die richtige Information gesprochen wird, wenn diese im entsprechenden Kanal und in der passenden Informationsgruppe gesendet wird, wobei auch darauf zu achten ist, daß sinnvolle Kurzinformationen entstehen. Die Einteilung der Meldungen in Informationskanäle und -gruppen wurde in Kapitel 3.5 erläutert.

Der Programmierer muß sich außerdem immer vor Augen halten, daß der Sender als Orientierungspunkt benutzt wird und daher bei Richtungsangaben der Standort des Benutzers maßgeblich ist.

Zur Erleichterung der Programmierung und um die Senderprogramme leicht lesbar zu machen wurden zwei Hilfsprozeduren (*tx\_id*, *txd*) geschrieben.

Beispielprogramm:

```

/* Sender bei Kreuzung Favoritenstr. */
#include "send.h"
const char    *p_str2 = "GUSSHAUSSCHTRASSE";
const char    *p_str3 = "TAUBSCHTUMMENGASSE ";
const char    *pos_str1 = "FAVORITENSCHTRAHSSE";
const char    *pos_str11 = "VIERTER BEZIRK.";
const char    *pos_str2 = " TECHNISCHE UNIVERSITET WIEN ELEKTROTECHNI-
SCHES INSTITUTSGEBEUDE.";
const char    *t_str1 = "UUU EINS ";
const char    *vers_str1 = "APOTHEKE";
const char    *zur_str = "ZUR";

void main(void)
{
    init();

    while(TRUE)
    {
        id.tid = 4;
        id.mid = 10;
        id.nbl = 1;
        id.bid = 1;

        chxor = chksum = 0;
        channel = ORIENTIERUNG;
    }
}

```

```

tx_id(channel);
txd(GTYP_1,T_INFO,PIC_KREUZUNG);
txd(GTYP_1,T_ASCII,pos_str1);
txd(GTYP_2,T_ASCII,pos_str11);
txd(GTYP_3,T_INFO,PIC_GERADEAUS);
txd(GTYP_3,T_ASCII,pos_str2);
txd(GTYP_3,T_INFO,PIC_EINGANG);
txd(GTYP_3,T_ASCII,pos_str1);
txd(GTYP_1,T_STOP,0);
delay(100);

chkxor = chksum = 0;
channel = VERKEHR;
tx_id(channel);
txd(GTYP_1,T_INFO,PIC_RECHTS);
txd(GTYP_1,T_ASCII,zur_str);
txd(GTYP_1,T_INFO,PIC_U_BAHN);
txd(GTYP_2,T_INFO,PIC_LINIE);
txd(GTYP_2,T_ASCII,t_str1);
txd(GTYP_3,T_INFO,PIC_HALTESTELLE);
txd(GTYP_3,T_ASCII,p_str3);
txd(GTYP_1,T_STOP,0);
delay(100);

chkxor = chksum = 0;
channel = AMPEL;
if (AMPELSTAND == ROT)
{
    id.mid = 1;
    tx_id(channel);
    txd(GTYP_1,T_ASCII,pos_str1);
    txd(GTYP_1,T_INFO,PIC_AMPEL_ROT);
    txd(GTYP_1,T_STOP,0);
}
else
{
    id.mid = 2;
    tx_id(channel);
    txd(GTYP_1,T_ASCII,pos_str1);
    txd(GTYP_1,T_INFO,PIC_AMPEL_GRUEN);
    txd(GTYP_1,T_STOP,0);
}
delay(100);
}
}

```

---

## Anhang A Codetabellen

### A1 Infogrammcodes

Bedeutung	Code
AMPEL	0
SCHUTZWEG	1
AUFZUG	2
ANKUNFT	3
KREUZUNG	4
HALTESTELLE	5
GLEIS	6
TUER	7
STRASSE	8
EINSTIEG	9
EINGANG	10
AUSSTIEG	11
AUSGANG	12
STIEGE	13
UNTERFUEHRUNG	14
ROLLTREPPE	15
GASSE	16
STOCK	17
ABFAHRT	18
EINBAHN	19
TOILETTE	20
WARTERAUM	21
WARTEHEUSCHEN	22
AMPEL_ROT	23
AMPEL_GRUEN	24
AMPEL_BLINKT	25
AMPEL_A_BETRIEB	26
RECHTS	27
LINKS	28

---

Bedeutung	Code	(Fortsetzung)
HINTEN	29	
VORNE	30	
HINUNTER	31	
HINAUF	32	
GERADEAUS	33	
SCHRAEG	34	
GEGENUEBER	35	
IST	36	
METER	37	
ZU	38	
VON	39	
NACH	40	
STRASSENBAHN	41	
BUS	42	
TAXI	43	
ZUG	44	
LINIE	45	
U_BAHN	46	
BAHN	47	
FLUG	48	
BAHNSTEIG	49	
FLUGHAFEN	50	
BAHNHOF	51	
HAFEN	52	
SCHIFF	53	
PARKPLATZ	54	
RESTAURANT	55	
HOTEL	56	
DAMEN	57	
HERREN	58	
POLIZEI	59	
VORSICHT	60	
HINDERNIS	61	



---

Bedeutung	Code	(Fortsetzung)
MARKIERUNG	62	
PAUSE	63	
INFORMATION	64	
UEBERFUEHRUNG	65	
TELEFON	66	
GELDWECHSEL	67	
KONTROLLE	68	
KARTENVERKAUF	69	
ROLLSTUHLGERECHT	70	
FAEHRE	71	
RAUCHER	72	
NICHTRAUCHER	73	
POSTAMT	74	
PASS	75	
ZOLL	76	
GEPAECK	77	
EINFAHREN	78	
UND	79	
ERDGESCHOSS	80	
KELLER	81	
HAUPTINGANG	82	
BEI	83	
NUMMER	84	

---

**A2 Lautschriftcodes**

Bedeutung	Code
Pause	0
a	1
a:	2
ae	3
ah	4
e	5
e:	6
ee	7
ee:	8
eh	9
eh:	10
i	11
i:	12
oo	13
oo:	14
o	15
o;	16
oe	17
oe:	18
u	19
u:	20
y	21
y:	22
ng	23
s	24
ss	25
schsch	26
sss	27
dd	28
v	29
w	30

---

Bedeutung	Code	(Fortsetzung)
z	31	
j	32	
f	33	
r	34	
p	35	
t	36	
k	37	
b	38	
d	39	
g	40	
l	41	
m	42	
n	43	
x	44	
h	45	

**A3 Kurzinformation**

```
const char kinfo_off[14] =
    {0x00,0xFF,0xFF,0x00,0xFF,0x00,0x00,0xFF,0xFF,0x00,0x00,0xFF,0x00,0xFF };
```

```
const char kinfo_on[7][14] =
    {
    { 0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 },
    { 0x00,0xFF,0xFF,0x00,0xFF,0x00,0x00,0xFF,0xFF,0x00,0x00,0xFF,0x00,0xFF },
    { 0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 },
    { 0x00,0xFF,0xFF,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 },
    { 0x00,0xFF,0xFF,0x00,0xFF,0x00,0x00,0xFF,0xFF,0x00,0x00,0xFF,0x00,0xFF },
    { 0x00,0xFF,0xFF,0x00,0xFF,0x00,0x00,0xFF,0xFF,0x00,0x00,0xFF,0x00,0xFF },
    { 0x00,0xFF,0xFF,0x00,0xFF,0x00,0x00,0xFF,0xFF,0x00,0x00,0xFF,0x00,0xFF }
    };
```

```
const char einkauf_vers[14] =
    { 0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 };
```

```
const char freiz_orient[14] =
    { 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00 };
```

```
const char freiz_vers[14] =
    { 0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 };
```

```
const char fortb_in_ggeb[14] =
    { 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF };
```

```
const char hilfe[14] =
    { 0x00,0x00,0xFF,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 };
```

## **Anhang B Flußdiagramme**

**B1 main**

**B2 dlt\_administration**

**B3 check\_kreisw**

**B4 choose\_message\_to\_speak**



**B5 speak\_msg**

**B6 chk\_key**

**B7 chk\_swt**

## Literaturverzeichnis

- [MAY92]     MAYER, P.: "I.R.I.S", Projektbericht, Institut f. Allg. Elektrotechnik u. Elektronik - Arbeitsgruppe Rehabilitationstechnik TU Wien, 1992
- [HIT86]     HITACHI : MICROCOMPUTER DATA BOOK 8/16-Bit Microprocessor, 1986
- [AUT89]     AUTHRIED; W.: "Sprachsynthesemodul", Diplomarbeit, TU Wien, 1989
- [KAU92]     KAUER, H.: "FM-Übertragungsstrecke", Diplomarbeit, TU Wien, 1992
- [KER90]     KERNIGHAN, B.W.: "Programmieren in C", 2.Ausg.  
Hanser Wien, München, 1990  
ISBN 3-446-15497-3