A MICROPROCESSOR-CONTROLLED, SPEECH-SYNTHESIS AID for BLIND SWITCHBOARD OPERATORS, TO BE FITTED TO MANUAL (20 + 100) SAPO SWITCHBOARDS.

Adriaan A. van Niekerk

Thesis submitted to the Department of Electrical and Electronic Engineering of the University of Cape Town in partial fulfilment of the credits for the degree of Master of Science in Applied Science, the rest of the credits being obtained as the Graduate Diploma of Engineering.

Cape Town

November, 1984.

ABSTRACT

The purpose of the investigation and hardware design was to provide a device or system that would enable a blind person to operate the standard manually-operated Floor Pattern Switchboard as supplied by Plessey South Africa to the SAPO (South African Post Office).

This type of Switch-board is a redesign, finished in 1981, of the old-fashioned type in use since the early 1950s. The major changes introduced were the replacement, by solid-state LED indicators, of the relatively expensive (due to the labour content in manufacture) and unreliable (due to the delicate mechanical assembly) shutter-type indicators used to signal incoming Exchange and Extension calls. The rotary dial was also replaced by an electronic push-button dialler with last number redial.

The device produced is microprocessor controlled and alerts the Blind Operator to switchboard activity by delivering voice-synthesized messages under control of a footswitch. The Blind Operator Aid programme is structured so as to allow flexibility of response to the various call types and the resulting operator actions required.

Particular attention was paid to making the Blind Operator Aid both inexpensive and capable of being retro-fitted to, and removed from, any of the Floor Pattern Switchboards already installed without damage. Modular construction allows easy field repair.

To date, 4 units have been built and are presently undergoing field trials to check the ergonomics, programme flow and general features.

A public relations exercise involving SATV, Radio and the Press will bring this product to the attention of the public and attempt to promote the employment of blind operators.

DEDICATION

To Mr Brian Suter for his help in the construction of the 4 field trial units.

To Plessey South Africa for the use of their facilities, for funding the prototype, and their patience.

To the SAPO for the use of their switchboard during development and for arranging the field trial.

To Mr Brian Redmile for proof-reading the draft.

To the SANCB (South African Council for The Blind) for playing "Guinea-Pig".

To Lolita Slabber for tracing all the diagrams.

To my family for tolerating a "loud-mouthed computer".

DECLARATION

I declare that this Thesis is my own unaided work. It is being submitted in partial fulfillment of the requirements for the degree of Master of Science in Applied Science to the University of Cape Town. It has not been submitted before for any degree or examination at this or any other university.

Adriaan Adam van Niekerk
(Name of canditate)

(Signature of candidate)

15th day of November, 1982

PREFACE

The device discussed herein was developed to overcome the inability of the blind operator to use a modern manual switchboard with LED indicators. The need was felt by the SAPO in 1980 after Plessey South Africa had redesigned the standard (20 + 100) manual Floor Pattern Switchboard, equipped at the time with mechanical shutter type indicators, replacing these with solid-state LED indicators.

The original request, to help the SANCB (South African National Council for The Blind), was to develop a model with protruding plunger-type tactile indicators. Unfortunately, the extension and exchange line circuits and indicators are mounted horizontally, this negating the use of the tactiles which are gravity released i.e. they are drawn up vertically by the current in the coil and allowed to fall down when de-energised.

The author became involved in 1983 due to the need for a thesis subject. This helped to make the project cheaper (commercial development projects are expensive, costing typically R35 per hour).

All the software development was done at the laboratory of Plessey Telecommunication Division on their microprocessor development systems, mostly after hours. The Object Code was debugged and tested at the author's home using a prototype scanner-cum-processor, with resident monitor, driving a speech synthesizer based on the National Semiconductor DIGITALKER module.

The actual layout of the PCBs, the PCB assembly and "nuts & bolts" of the 4 field trial units were willingly done by Mr. Brian Suter as part of his 2-year Graduate Trainee period under the author, who thanks him for his efforts. He also ran basic tests on some of the software routines after they had been written by the author.

CONTENTS

CONTENTS continued

Chapter 1

Introduction

1.1 Why a Blind Operator Switchboard ?

1.2 Overview of Switchboard Operation

- 1.2.1 Front Panel Controls
- 1.2.2 Shelf Controls
- 1.2.3 Operation
- 1.2.4 Conclusion
- 1.2.5 Modification for Blind Operation

1.3 What is Done Elsewhere ?

1.1 Why a Blind Operator Console?

For many years the Pioneer School for The Blind at Worcester, controlled by the National Education Department, has trained both completely and partially blind switchboard operators. In 1984 there were 7 on the course, but as many as 12 have been handled per annum. This work is performed using adapted manual switchboards fitted with tactile indicators.

The ready acceptance of the blind operator by commerce mandates that the cost and any possible inconvenience to the firm is minimal. Thus the switchboard installation and rental fees must equal that of the standard units i.e. the cost of any adaptation must be borne by the SAPO and should ideally be only for a small retrofit task.

Many of the firms large enough to dedicate one person to switchboard operation (and not also making tea and typing letters) need more than 25 extensions and 5 exchange lines. This reguires the installation of a manually operated floor-mounted switchboard, with a maximum capacity of typically 20 exchange lines and 100 extension lines. This is one of a standard range of units supplied, installed and maintained by the SAPO.

1.2 Overview of switchboard operation

For the reader who may be unfamiliar with manual switchboards, a short description of the controls and operating procedures is given below to aid in understanding the reason behind the decisions taken as regards the design of the Blind Operator Aid. Fig.1.1 overleaf refers. The front view of the LED Floor Pattern Switchboard is given in the upper half, and the plan view of the control shelf in the lower half.

The (20 + 100) Floor Pattern Switchboard console is 1,2m high, 0,7m wide and 0,4m deep. A horizontal, 0,4m deep shelf, situated at elbow height for a seated person, protrudes towards the operator.

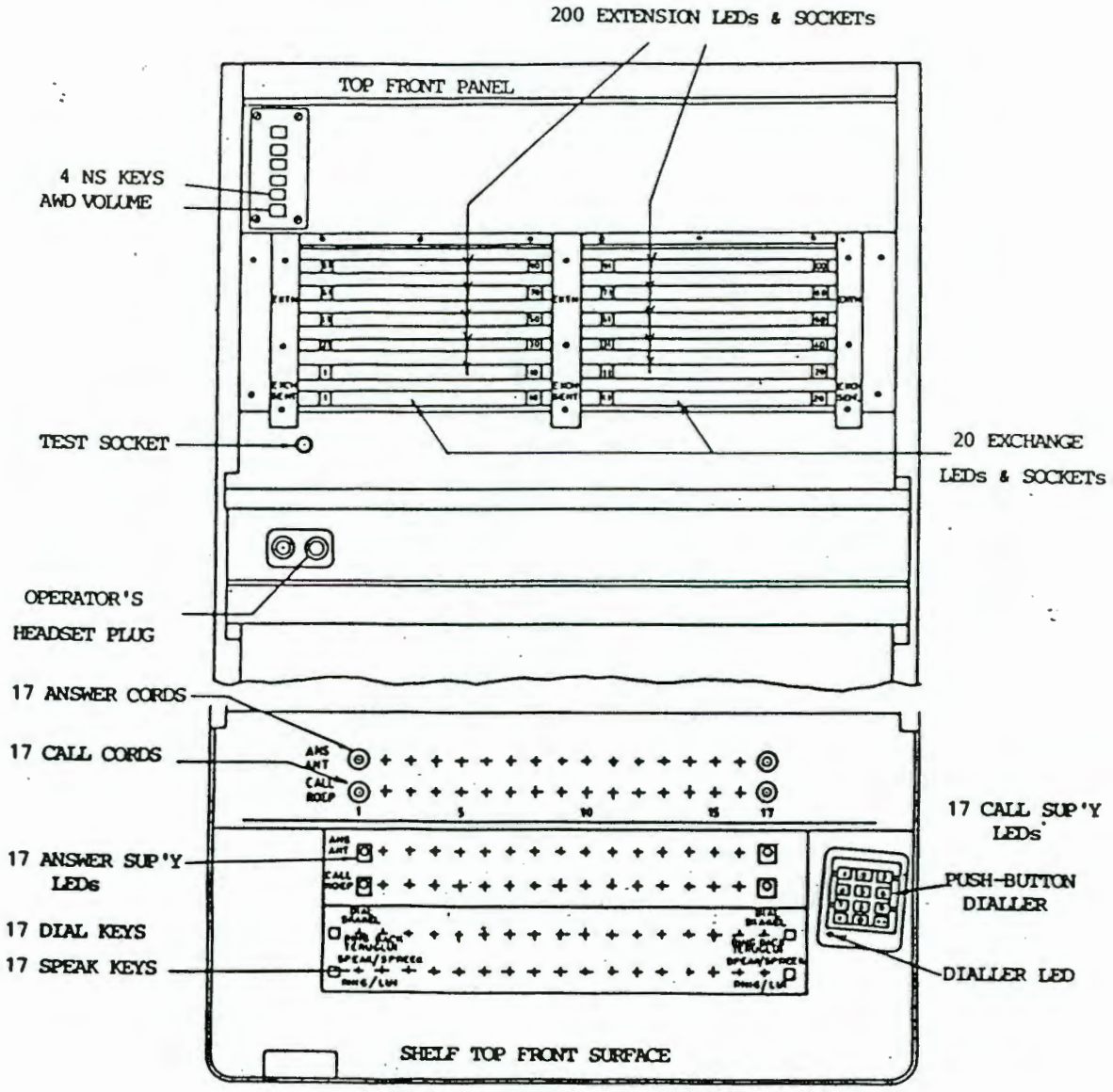


Fig.1.1 PICTORIAL VIEW OF (20 + 100) FPS

1.2.1 Front Panel Controls

On the front panel of the console are rows of LED indicators, grouped in sets of 10. Below each LED is an associated socket. 20 LED/Socket pairs are linked to incoming exchange lines and 100 LED/Socket pairs are linked to extension telephones.

The physical connections to the wires in the building are in the rear of the console and do not concern the operator. The LEDs are driven by the circuitry, (described in Appendix A.1 and A.2), each LED lighting in response to activity on the associated extension or exchange line.

There is also a test socket (discussed later), a 3-position volume switch for an AWD (audible warning device) and 4 NS (Night Service) key-switches.

The AWD is an electronic tone-sounder that alerts the operator to some switchboard activity, such as an exchange line ringing, a telephone going off-hook, or a call terminating. The sounder is housed in the shelf.

The NS keys, when operated, connect 20 selected extensions (5 per key) directly to the 20 exchange lines in a pre-determined manner.The selection is done with cross-wiring in the rear of the console by the SAPO technician at time of installation. When the NS keys are in the day position, the extensions are connected to the normal circuitry.

1.2.2 Shelf Controls

The shelf is actually a hollow box, about 75 mm high, with a hinged lid. In the base are various circuit elements associated with the controls in the lid.

There are 17 identical sets of controls in the lid of the shelf, with sets grouped next to one another, left to right. Each set, or "Cord Circuit", is concerned with one conversation and comprises 2 jack-plugs/cords, 2 LEDs and 2 Keys (i.e. switches), grouped from the rear to the front of the shelf.

1) Answer Cord: This is a 6mm diameter jack-plug, attached to a braided 3-wire cord which hangs down inside the console base in a loop. A weighted pulley returns the cord into the console base when the jack-plug is released from a socket. If the plug is inserted into an appropriate extension or exchange socket the operator may, by operating one of the keys, speak to the calling extension or answer the ringing exchange line.

2) Call Cord: This is similar to the Answer Cord, except that the operator may ring a desired extension, or sieze and dial out on a selected exchange line, by operating appropriate keys.

NOTE.

If two extensions, or an extension and an exchange line, are associated with the Answer and Call Cords in any set, then a full conversation is set up, the operator being capable of speaking to, and hearing, both parties if the associated key is operated.

3) Answer Supervisory LED Indicator: The Answer Sup'y LED is lit if the Answer Cord is plugged into an extension socket, with the telephone off-hook. Bouncing the cradle switch makes the LED flash in sympathy.

   If the Answer Cord is plugged into a ringing exchange line socket, then the LED is lit after the operator answers the call using the Speak key, remaining lit until the Cord is removed, or, if extended to an extension, until after the extension has gone on-hook i.e. hung-up. The incoming call on the Exchange Line has no influence on the LED since the on-hook signal is not transmitted to the Switchboard by the Exchange.

4) Call Supervisory LED Indicator: The Call Sup'y LED is similar to the Answer Sup'y LED except that it is associated with the Call Cord side of the Loop.

5) Dial/Ring-back Key: This is a 3-position, centre-normal key, "Dial" in forward position, "Ring-back" in back position.

   a) Dial: The push-button keypad, also situated on the shelf, is connected to the exchange line associated with the Call Cord by means of the Dial key, allowing the operator to dial the desired Exchange numbers. The Speak key is needed only to allow the operator to monitor the progress of the call to the Exchange. An LED on the keypad pulses during dialling.

   b) Ring-back: The extension telephone associated with the Answer Cord is rung while the key is operated.

6) Speak/Ring Key: This is a 3-position, centre-normal key-switch, "Speak" in forward position, "Ring" in back position.

   a) Speak: The operator's handset is connected to the centre point of the Answer and Call Cords, allowing speech to both parties.

   b) Ring: The extension telephone associated with the Call Cord is rung while the key is operated.

In the front of the shelf are two sockets for the operator's handset or headset, the first for normal use, the second to allow another operator to listen for training purposes.

7) Test Socket: This lets the operator test the Answer and Call Cords by plugging in and seeing the Answer or Call Sup'y LED light.

1.2.3 Operation

Having detailed the purpose of each of the switchboard controls, it is now possible to discuss various call types. Assume for clarity in the descriptions that the two Extensions involved are #1 and #2, the Exchange line is #3 and the Cord Circuit is #4. The actual range of Extension numbers lies between 1 and 100, those of Exchange lines between 1 and 20 and Cord Circuits between 1 and 17.

a) Extension to Extension Call

1) Extension #1 lifts the telephone and extension LED #1 lights. The AWD gives a tone to warn the operator.

2) The operator selects the Answer Cord #4 and plugs it into extension jack-socket #1, causing the Extension LED to go out and #4 Answer Sup'y LED to light. By operating Speak key #4, the operator can speak to the calling party.

3) The operator now plugs the Call Cord #4 into extension socket #2 and rings the telephone by operating Ring key #4 briefly.

4) The operator may retire from the call, either before or after the extension answers, by returning the Speak key to centre-normal. When extension #2 answers, Call Sup'y LED #4 lights.

5) At the end of the call either, or both, of the extensions replace their telephone handsets, causing the corresponding Sup'y LED to go out. The AWD emits a tone to warn the operator to remove the Cords, or to speak to the remaining extension and process another call type.

b) Exchange line to Extension call

1) The incoming exchange ring-current causes exchange LED #3 to flash in time to the cadence, as well as the AWD to give a cadenced tone.

2) The operator answers the call by inserting Answer Cord #4 into the jack-socket and operating the Speak key. The Answer Sup'y LED lights and the operator speaks to the outside caller.

3) Thereafter, the operation is as in a)3, 4 and 5, except that both Sup'y LEDs go out when the extension replaces the handset.

4) If the extension wishes the call to be transferred to another extension, the handset cradle switch may be depressed and released, "flashing" the operator, several times. The AWD will sound to warn the operator and both Sup'y LEDs will flash. The call is not dropped in the exchange due to it being an incoming call.

c) Extension to Exchange Call

1) Operations a) 1 and 2 occur.

2) The operator may plug the Call Cord into the exchange line socket and retire from the call, in which case the extension may dial the number after hearing dial tone.

OR

3) The operator may sieze an exchange line using the Call Cord and Speak key. After hearing dial tone, the Dial key is operated, connecting the dial across the exchange line. The operator dials the number, returning the Dial key to centre-normal to hear the distant end when dialling is completed as indicated by

1.2.4 Conclusion.

The entire task of supervising the switchboard is dependent upon monitoring the behaviour of:-

a) 100 Extension LEDs.

b) 20 Exchange LEDs.

c) 34 Supervisory LEDs.....

.....coupled with the position of 34 Cords. The position of the cords may be deduced from the sequence of LEDs lighting and extinguishing.

1.2.5 Modification for Blind Operation.

The adaption of these manual switchboards for use by the blind has several problems:-

a) Cost In 1980, 12 were adapted at a total cost of R15 800 and in 1981, the last 15 available at a total of R19 700.

b) Availability The Post Office Exchange alerts a subscriber to an incoming call by putting an AC voltage onto the line to ring the telephone or PABX. This ringing signal (approximately 90V) is applied at a rate of (0,5s On/0,5s Off/0,5s On/2s Off), and so on until the call is answered. The most suitable tactile indicator used to detect this type of signal is the Drop-flap type which gives a permanent indication of an intermittent electric current after it is first detected (see Appendix). This latching action is needed to avoid the blind person having to pause for up to 3 seconds at each indicator, waiting for exchange line ringing, thus leading to slow reaction by the operator.

The Eyeball indicator (see Appendix) is the most suitable to signal a steady state current such as flows when a telephone is lifted.

These indicators and the large housings needed for them are no longer manufactured due to cost and aesthetic reasons.

To summarise, a device was needed that would enable a blind operator to service all calls on the switchboard, rapidly and with few errors. The cost and inconvenience to the firm should be minimal, with little disruption to service as the device is installed or removed. Damage to the switchboard should be zero. The tactile modifications to earlier switchboards were so extensive that the whole switchboard had to be removed each time.

1.3 What is Done Elsewhere ?

Letters were written to the Royal Societies for The Blind in the U.K and Australia. A tactile switchboard manual from Tadiran Electronic Industries in Israel was available.

The Australian Society replied with a letter and brochures on two devices:

a) The Tallyphone: This is really a computerised staff record-storage machine, being capable of tabulating department, telephone number etc., but not having any application for a manual switchboard. Cost A$8 000. The Tallyphone has an integral screen-to-speech converter so that a blind operator could act as a telephone directory for a large firm with upwards of 800 telephones.

b) A Light Probe: This is a device that responds to the light emitted by the LEDs on a switchboard. Sensitivity is adjustable, with the sensor housed in a pen-shaped probe. The control box houses a loudspeaker which emits an adjustable tone when a scanned LED is lit. This type of probe is already used in South Africa by partially sighted

operators on the SAPO (5 + 20) switchboard which is controlled by miniature key-switches and thus has an uncluttered front panel. It is not suitable for use on the Floor Pattern Switchboard which usually has several Cords plugged in, making it difficult and time consuming for the operator to scan the 120 LEDs.

The Australian Society has no knowledge of synthesized speech manual-switchboard aids.

The Society in the U.K. did not reply.

Tadiran Electronics Industries in Israel provided a brochure

Tadiran Electronics Industries manufacture a modified upright Floor Pattern Switchboard. A loudspeaker emits an 800 Hz tone when an exchange line rings, and a 400 Hz tone when an extension lifts the telephone, a modulated tone sounding when the two occur together. The operator uses a jack-plug to scan the sockets until the tone alters or stops, upon which he plugs in to answer the call. There are keys to control speech or ringing to the extensions and dialling on the exchange lines.

This board has a maximum capacity of 10 Exchange Lines and 50 Extension Lines (referred to as:- (10 + 50)) with 10 simultaneous conversations. This is not suitable for South African requirements, which are (20 + 100) with 17 simultaneous conversations.

In view of the lack of suitable devices or information, it was decided by SAPO to proceed with a South African designed device to maintain compatability with the 1500 new LED type Floor Pattern Switchboards installed since 1982.

Chapter 2

Design Concepts

2.1 General Philosophy

2.2 Input From Switchboard

2.3 Output To Blind Operator

2.4 Blind Operator Aid Control

2.5 Sanity Control

2.6 Power Supply

Design Concepts

Having considered, in Chapter 1, the operation, controls and features of a switchboard, as well as the difficulties facing a blind operator, it is possible to examine the various sections of a Blind Operator Aid.

2.1 General Philosophy

From the outset, microprocessor control of the Blind Operator Aid was a fundamental tenet. The author's previous experience with devices of this complexity (e.g. recording rain-gauge, computerised telephone directory, cassette data-tape reader), indicated that the ergonomics, features and internal functions of the final model usually vary sufficiently in detail from the initial field-trial models to warrant a software-controlled machine.

Provided that the interfaces to the external environment were carefully chosen to cover all aspects possible, and the internal hardware had reasonable excess capacity to cover some future enhancements, a flexible, microprocessor-controlled device seemed a logical choice.

2.2 Input from Switchboard

A method of inputting the status of each of the 154 LEDs to the Blind Operator Aid was needed. The simultaneous reading of 154 LED voltages is hardware intensive and thus complex and expensive. A Scanner Interface was selected, allowing the state of each LED to be monitored at regular intervals. Decisions are taken on sequential states of each LED. These are correlated over suitable intervals to determine the time "ON", "OFF", and hence possible "FLASHING".

A digital multiplexer is used, since the Blind Operator Aid is microprocessor controlled for flexibility of operation. This implies that the circuit must have at least the following:-

a) An address range of 1 to 154 inputs

b) A common output to the microprocessor, at TTL level, to communicate the instantaneous status of the addressed LED, and hence enable the programme to decide whether "LED ON" or "LED OFF" or "LED FLASHING" etc. is applicable.

Only two methods of LED sensing are practical:-

a) Optical: The easiest retrofit to a switchboard is possible when the technician merely has to screw on some form of adaptor. Thus an external device that sensed the LEDs optically was the first method considered. Fig.2.1 refers.

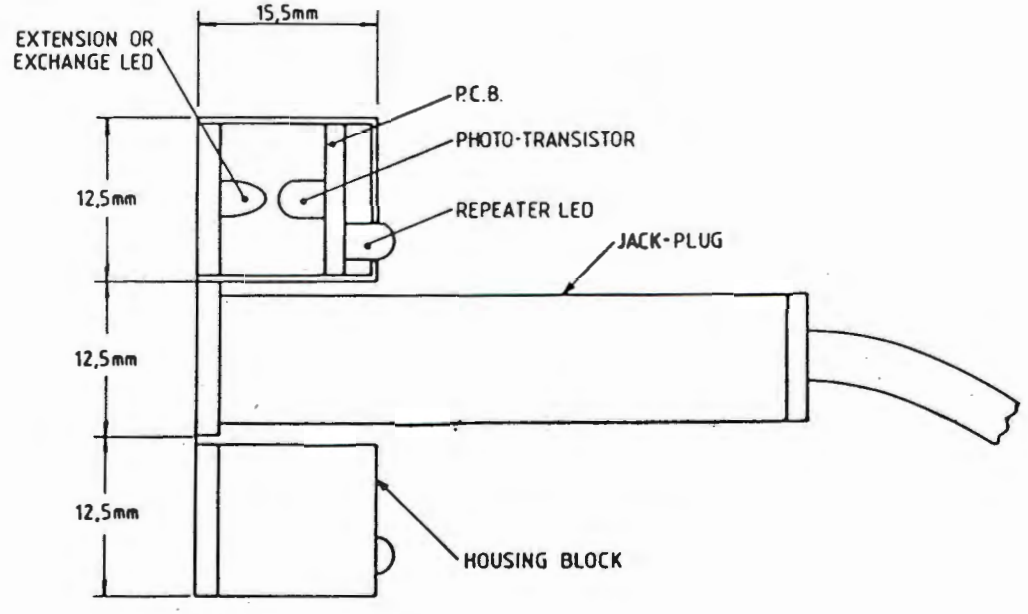


Fig.2.1

Miniature photo-transistors, mounted on the rear side of a PCB faced the LEDs. However, since the LEDs were now obscured, repeater LEDs were mounted on the front of the PCB, spaced so that they coincided vertically with the jack-sockets.

Miniature photo-transistors available from Siemens were 2mm in diameter and 4mm high. The repeater LEDs were the same size and emitted a diffused red light.

This scheme was rejected, before any major circuit design was completed, for the following reasons:-

1) Cost: The price of the photo-transistors and LEDs was R100, plus that of the housings estimated at R110 total plus the PCBs at R50 total.

2) Not Fail-Safe: Due to the fact that the external LEDs repeated the states of the actual LEDs, any power failure to the repeater LEDs or malfunction of the extra circuitry needed would prevent a sighted operator from using the switchboard.

3) Space: The area of PCB needed to place the photo-transistors on one side and the LEDs on the other, as well as that needed to run the tracks to the PCB connector, left only 12mm between sensor units for the operator to insert the jack-plug.

   This proved to be an irritation in simulated tests, slowing down the operator.

b) Electrical

The drive voltage to, or the current through, the indicator LED was next considered as a stimulus.

The LED drive circuits of the extension and exchange indicators (Ref. A.1 and A.2) are similar, both having a polarity opposite to that of the Supervisory Indicator LED (Ref. A.3). In addition, the two former are similar in PCB construction i.e. in units of 10, while the Supervisory Indicators are individual units with separate connector plugs.

It is easy to gain access, via ribbon cables, to the LED leads on the extension and exchange units, allowing the

voltage change across the LEDs to be sensed. Since the normal operation of the LEDs should not be affected, the sensors should have a high impedance relative to the LED driving circuits.

The individual plugs supplying the Sup'y units are easy to remove and connect across other plugs, allowing a current-sensing resistor (with a value low enough not to affect the normal operation of the LEDs) to be placed in series with the leads. This current-sensing resistor can be used to drive a voltage-gain transistor (conveniently inverted) to provide an output similar to that on the Extension and Exchange Indicator units.

The above have been used in the design of the Blind Operator Aid. The 154 input interfaces to the scanner are merely resistors from each LED, the scanner being referenced to the switchboard OV. The 34 Sup'y LEDs each have an additional interface of a resistor in series with the Indicator supply, driving an inverting transistor, the cost penalty being small.

The actual design, discussed in Chapter 4, is a scanner based on CMOS digital-multiplexer integrated circuits, with suitable resistors and protective diodes.

2.3 Output to Operator

Two methods of outputting information regarding call status to the Blind Operator were considered:-

a) Tactile: The orientation difficulties with protruding plunger tactiles, the slow response to exchange calls when using non-latching tactiles, (discussed previously), and the need to monitor the Cord circuits continuously to make sure a called party has answered, (when extending and then retiring from calls on a busy switchboard), make the use of tactiles unattractive.

b) Audible

Again, two means of providing audible messages were considered:-

1) Tones: A series of high, low, modulated etc. tones can be given to indicate call status as in the switchboard from Tadiran (Para. 1.2). This method may be adequate when used in their unit, where 60 jack-sockets in one straight line from left to right are scanned with a plug, but becomes unwieldy in the SAPO switchboard where 120 jack-sockets are arranged in 12 units of 10 each. Modification of the basic Floor Pattern Switchboard to accommodate the Tadiran arrangement was not even considered due to cost, time scales and incompatibility with existing equipment.

Variations such as coded tone pulses i.e. up to 9 high tone bursts for the most significant digit etc. were rejected as requiring too great a concentration from the operator, as well as slowing down operators due to the average message time being 5 seconds for 10 bursts of tone (0,5 second between bursts was found adequate).

2) Speech Messages such as "E-one" or "E-fifty-six" and "Cancel-L-three" are unambiguous, concise and rapid.

In addition, it is desirable to generate other types of message, such as "Time-is-four-thirty" to provide a Time Clock, or "Danger" to signal a unit malfunction.

There are various methods of synthesizing the speech, these being discussed in Chapter 4. Speech synthesis itself was chosen as the output method due to the flexibility and ease of comprehension.

2.4 Blind Operator Aid Control

The choice here was a simple one. A standard commercial PCB, based on the 8085 microprocessor, containing ROM, RAM, CPU clock and an I/O peripheral device is used by SAPO, who requested that it be utilised to simplify their field maintenance.

The major facilities offered by the CPU, based on the SABUS, are:-

a) Interrupts: These allow time-related events such as a 100ms timer or a programme sanity reset to be serviced.

b) Reset Button: This allows the CPU to be restarted from the begining of the programme if needed.

c) 8K ROM: The ROM stores the operating programme for the CPU. This ROM size seemed adequate for a device such as the Blind Operator Aid.

d) 1K RAM: This allows storage of temporary variables such as Real Time, LED status as a call progresses etc. Since the number of Extensions plus Exchange lines plus Sup'y pairs is 137, (hence possible messages 137), it was felt that this storage length was adequate.

e) 24 I/O Lines: These allow external interfaces to be read and controlled. 154 LEDs can be addressed by 8 lines (2^8 = 256 | 154). 8 Lines can be used to input control switches and the multiplexed LED status. The remaining 8 lines can be used to control output devices such as the voice synthesizer. Thus, the size of I/O seemed adequate.

Considering the above, it was decided to use the suggested module.

2.5 Sanity Control

From the author's previous experience of microprocessor-controlled devices, it was considered necessary to provide some form of sanity check on the programme that will override the normal operation of the device. Some event, such as a random alteration of RAM contents due to a mains-borne voltage spike, or a low-voltage "brown-out", may cause corruption of a subroutine return address or a limit variable in a programme loop. Unless corrected externally, it is unlikely that the device will resume normal operation. The sanity control must interrupt the microprocessor and restart the programme at a convenient point.

It was decided to use a counter chain, driven from a Clock circuit, (independent timing-pulse source), to provide a TRAP interrupt to the 8085 microprocessor. The TRAP input cannot be ignored by the 8085. This concept is referred to as the "Watchdog" for obvious reasons ! The Watchdog is restarted regularly by the normal running of the programme, the time between restarts being less than the trigger period of the Watchdog.

2.6 Power Supply

The switchboard operates from either a mains-driven supply or from 24V batteries and charger.

It was decided to power the Blind Operator Aid from a built-in mains-driven supply for the field trial models, as this would show any problems encountered due to brown-outs or mains spikes, as well as testing the operation for a sighted operator during mains-fail conditions. The PSU (Power Supply Unit) should be implemented so that it can be replaced by a 24V DC-driven PSU with no alteration other than the provision of the alternate unit.

The metalwork of the mains PSU is required by a SAPO regulation to be connected to the mains earth. The DC output of the PSU is connected to a separate earth for safety.

Chapter 3

The Working Specification

3.1.0 Scope

3.2.0 Construction

3.3.0 Hardware

3.4.0 Software

3.5.0 Operation

The Working Specification

Examination of the various options discussed in the previous chapter led to the following main parameters:

a) The sensors should be as cheap as possible, preferably avoiding any extra interface such as opto-couplers and repeater LEDs, which add significantly to the cost.
b) The unit should be microprocessor-controlled for easy and rapid changes to the ergonomics.
c) The entire unit should be an easy retrofit to existing installations.
d) The output to the operator should be a speech synthesizer, with short messages.
e) The speech output should be easily altered if the field trial showed a need.

After a round-table discussion between the SAPO, the SANCB and the author, the following guidelines were drawn up in the form of a working specification:-

PLESSEY

TELECOMMUNICATIONS DIVISION

SPECIFICATION

SPECIFICATION Nº
PROVISIONAL

A MICROPROCESSOR-CONTROLLED, SPEECH-SYNTHESIS AID
for BLIND SWITCHBOARD OPERATORS, TO BE FITTED
TO MANUAL (20 + 100) SAPO SWITCHBOARDS

ISSUE STATE

This Issue consists of the following Pages:

Page Nº	Issue
1	A
2	A
3	A
4	A
5	A
6	A

Page Nº	Issue

Page Nº	Issue

Prepared : A.A.VAN NIEKERK ENG.

Approved : B.D. REDMILE CDET

Approved : A A VAN NIEKERK MRDT

Approved :

Approved :

Approved :

Issue Nº						
Date						
Change Note Nº						
Issue Nº						
Date						
Change Note Nº						

1.0 SCOPE

1.1 This Specification describes a device with voice output to be used with floor pattern switchboard type 1-104-10328-XXX to aid blind operators. The device can be fitted to the switchboard on the subscriber's premises, does not alter the operating procedures for a sighted person, and can be removed at any time leaving the switchboard in a usable state.

1.2 The device is connected to the Exchange, Extension and Supervisory indicators of the switchboard. It contains a speech synthesizer and announces changes which require the attention of the operator. The device is microprocessor based and modular.

2.0 CONSTRUCTION

2.3 The device shall consist of a free standing case, suitable for placement on top of the switchboard.

2.2 The device shall be mains powered. A connector of the IEC 320 type shall be fitted in the case.

2.3 The device shall be connected to the switchboard by flat ribbon cables brought down the rear of the switchboard. The cables shall enter the switchboard through a gap between the door and the frame.

2.4 The case shall include a loudspeaker and volume control. The levels, impedance and possible method of coupling to the operator's headset may be specified later.

2.5 A Footswitch or similar device shall be provided and shall be connected to the device by a suitable plug and socket. The switch will be used by the operator to control the delivery of messages.

3.0 HARDWARE

3.1 The circuit shall be designed on a modular basis. The speech synthesizer, input interfaces and microprocessor shall be on separate printed circuit cards using standard modules and busses where possible.

3.2 The unit shall be connected to the exchange and extension indicators via dil ic sockets on the indicator PCBs if PCB area permits.

3.3 The unit shall be connected to the supervisory indicators by means of Molex connectors which mate with the connectors on the flying lead (terminals E and F). One of the 34 (30 on 10 + 50 switchboards) sockets can be used to provide connections to tag 2 of the pilot circuit (Batt) and Earth.

3.4 The input interface for the exchange and extension indicators shall respond to the voltages occurring at the point of connection. The interface shall not draw appreciable current through the LED which may light and give a false indication to a sighted operator. The indicators shall continue to operate normally if the mains supply to the control unit is off.

3.5 The input interface for the supervisory indicators shall monitor the current drawn by terminal F of each indicator individually. The indicators and pilot circuit shall continue to operate normally if the mains supply to the control unit is off or the ribbon cable is disconnected from the control unit. The latter condition means that part of the interface circuit will have to be mounted in the switchboard.

4.0 SOFTWARE

4.1 When power is applied to the unit it shall check the EPROM and RAM memory and announce the result with a suitable voice or tone message.

4.2 The unit shall ignore indications (on or off) lasting less than 0,1 second to prevent spurious noise on a line from giving repetitive irritating messages. If necessary the delay shall be longer on supervisory indicators so that the unit does not respond to dial pulses.

4.3 The unit shall recognise that an exchange line has stopped ringing or been answered when three seconds elapse during which the indication is off.

4.4 The unit shall recognise that an exchange line is on hold or that an extension has not answered if one of the supervisory indicators is off while the other is on for 30 seconds.

4.5 The unit shall maintain First-In-First-Out queues for the various types of message. The structure of the software shall be such that additional types (up to a total of at least 8) can be added if necessary. Initially the following types are required:

4.5.1 Clear Cord Circuit
4.5.2 No Answer
4.5.3 Exchange Line
4.5.4 Extension Line
4.5.5 Time of Day

NOTE: The type names given above and the corresponding words said by the synthesizer will be fixed later.

4.6 The queues shall be allocated priorities as indicated above. Messages shall be taken from the highest priority queue with messages waiting. Messages which are no longer valid shall be deleted.

5.0 OPERATION

5.1 A short burst of tone may be given each time a message is added to a queue.

5.2 The operator shall have control of the rate of delivery of messages. During the field trial different modes of operation shall be evaluated e.g.

a) deliver one message per operation of the switch,
b) pause while switch depressed.

5.3 A message shall be repeated until the condition causing it has been removed or it has been repeated the maximum number of times. Messages repeated the maximum number of times shall be placed at the back of the queue.

5.4 The 30 second timeout for no answer shall be restarted when the message is announced, i.e. the message shall not be put back in the queue until another 30 seconds have passed.

END

Design Details

4.1 General

The contents of this chapter discuss the detailed workings that preceded the final circuit configuration. Portions of the Blind Operator Aid and Switchboard circuits are reproduced in detail and block form as needed.

4.2 Switchboard/Scanner Interface

Having decided in 2.2.(b) that the interface between the Indicator LEDs and the Scanner input should be a low cost resistive type, the arrangement shown in Fig.4.1 below was proposed. The Indicators, Pilot Circuit and Interface (discussed overleaf) were analysed and the design calculated to ensure that all Blind Operator Aids and all Switchboards are guaranteed to work together. It was essential to design the circuits for operation over the full ambient working temperature range and over the worst case component tolerances encountered in mass production, since it is neither practical nor economical to match units such as an old installed Switchboard and a new Blind Operator Aid.

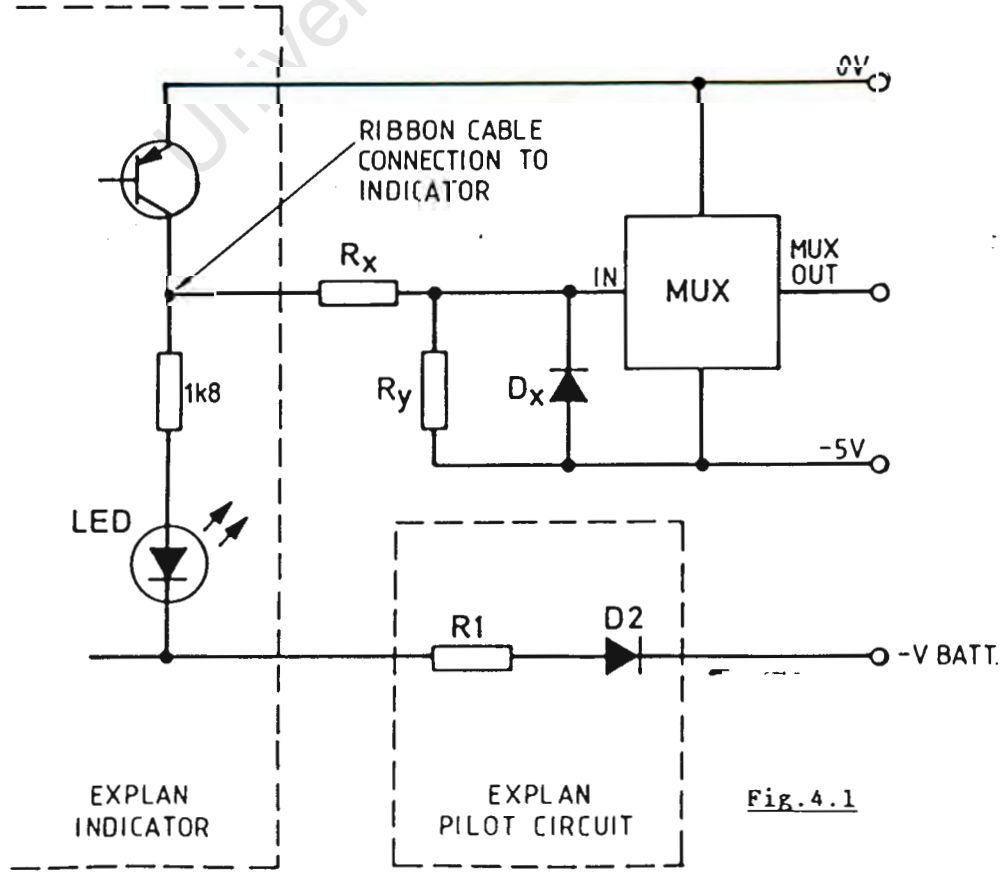


Fig.4.1

4.2.1 Extension & Exchange Indicator Interface

Refering to Fig.4.1 on previous page:-

1) The LED, 1K8 resistor and Transistor in "EXPLAN. INDICATOR" are those already existing in the Extension and Exchange Indicators (A.1.1 and A.2.1).

2) R1 and D2 in "EXPLAN. PILOT CIRCUIT" are those already existing in the Pilot Circuit (A.4.1).

3) R_X and R_Y are the Interface resistors to the Scanner Multiplexer, one each per input.

4) Diodes D_X protect the Multiplexer inputs against voltages more negative than -5V on the left-hand end of resistors R_X, one diode per input.

5) The connection between R_X and the Indicator is by means of a ribbon-cable plug and socket on the Blind Operator Aid Interface PCB, and a soldered connection onto the Indicator PCB at the other end.

This arrangement converts the relatively large analogue voltages (approx. 28V change) appearing at the output of the Indicator to logic "1"s and "0"s, for the digital Multiplexer to scan, as follows:-

1) With the Indicator transistor ON, R_X is connected to 0V. This is seen as a logic "1" by the Multiplexer input.

2) With the Indicator transistor OFF, the LED pulls R_X towards $-V_{bat}$ via the 1K8 resistor. The voltage on the Multiplexer input is clamped at -0,6V w.r.t. the -5V by D_X. This is seen as a logic "0" by the Multiplexer input.

3) With R_X not connected to an Indicator, R_Y provides a logic "0" on the Multiplexer input i.e. any Indicator not connected is seen as OFF. This is convenient for the software, since no special action need be taken for use on

under-equipped Switchboards i.e. any Indicators not fitted are merely sensed as being permanently OFF.

The factors considered in the Interface design are discussed below. Fig.4.1 should be considered in conjunction with the individual diagrams mentioned:-

a) All 120 Extension and Exchange Indicators in the Switchboard are directly connected to R1 in the Pilot Circuit (A.4.1). This allows any LED that is ON to sound the AWD as discussed in (A.4).

b) The parallel Indicator connection in a) above has the result that for each OFF LED the current from Interface resistor R_X, (Fig.4.1), will flow through the 1K8 resistor and the LED into R1 of the Pilot Circuit, possibly affecting the operation of the Pilot Circuit. The total "leakage" current will have a maximum value when all LEDs are OFF, V_{bat} is at the maximum of -28V and the -5V rail of the Blind Operator Aid is at 0V. This last condition could occur if the mains supply to the PSU of the Aid fails or if the PSU itself goes faulty. The relatively low impedance of the Blind Operator Aid circuitry, when off, will pull the -5V rail very close to 0V.

c) The Interface leakage currents flowing through R1, b) above, point to R1 having as <u>low</u> a value as possible to minimise the effect on the Pilot Circuit. However, R1 must still have a value <u>high</u> enough to turn on TR1, (A.4.1), when the <u>minimum</u> current from any Indicator LED flows through R1.

The procedure for finding suitable values for R1, R_X, R_Y, as well as the type of Multiplexer to be used, is given below:-

a) The minimum current from any LED indicator is deduced by analysing the circuit as shown in the Appendix. This current is a major consideration, since, with many thousands of Indicators already installed in the field, it is not possible

to alter the design of the Indicator to provide a different current, when ON, to suit the design of the Interface.

$I_{LED\ min}$ was found to be 5,7mA .....(1)

b) The maximum voltage required to turn on TR1 in the Pilot Circuit, (A.4.1), is next analysed as shown in the Appendix. This voltage, in conjunction with $I_{LED\ min}$, sets the minimum value of R1 that is guaranteed to turn on TR1 when the current from only one Indicator flows through R1.

$VR1_{max}$ was found to be 660mV .....(2)

A minimum value for R1 in the Pilot Circuit can now be found:-

$$R1_{min} = VR1_{max} / I_{LED\ min}$$
$$= 116\Omega$$

Take R1 = 120Ω, ±2% as the next higher standard value .....(3)

NOTE: R1 is 220Ω in the standard production Pilot Circuit and will need to be changed when installing the Blind Operator Aid. It can be left as 120Ω if the Aid is removed.

c) Taking the value of R1 deduced in (3) above, it is possible to get a very rough approximation of the magnitude of resistors R_X, (Fig.4.1), in order to choose between LSTTL and CMOS technology for the Multiplexer. From the data sheets for TR1, (Pilot Circuit A.4.1), all transistor samples are guaranteed to be off with 400mV across the base/emitter junction over the ambient temperature range of the Switchboard. In the calculation below, it is assumed that V_{bat} is the maximum of 28V specified for the Switchboard and that the Blind Operator Aid PSU is off, thus pulling R_X up to 0V via D_X. No account is yet taken of V_{LED} or the 1K8 resistor in the Indicator circuit. Thus:-

$$R_{X\ approx} = (V_{bat\ max}/(0,4/R1))*120$$

$$= 1,7M\Omega \text{ .....(4)}$$

The factor 120 accounts for the 120 parallel Indicators and Interface resistors R_X.

d) The value of R_X deduced in (4) above precludes the use of an LSTTL Scanner/Multiplexer which would require R_X to be of the order of Kilohms. Thus, an analysis of resistors for a CMOS digital Multiplexer is performed, as shown in the Appendix, to deduce values for R_X and R_Y. The actual values calculated are then checked, as shown below, to see if they are suitable using the value of R1 calculated in (3) above and the criterion that a voltage less than 400mV should be produced across R1 in order not to sound the AWD when no LEDs are ON.

From the Multiplexer analysis in the Appendix:-

$$R_X = 1{,}2M\Omega \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

It is now possible to calculate accurately the effect of the Interface leakage on R1 in the Pilot Circuit, taking into consideration the 1K8 resistor and V_{LED}. Since there are 120 resistors R_X etc., the equivalent circuit for determining VR1 is given in Fig.4.2 overleaf. The maximum voltage across R1 will occur if the Blind Operator Aid power supply is off, V_{bat} is a maximum, R_X has the lowest and R1 the highest tolerances.

If $V_{R1} <<< V_{bat}$, then:-

$$V_{R1} = ((V_{bat\ max} - 2V_{be} - V_{LED\ min})/(R_{X\ min} + 1800 + R1)) * R1_{max} * 120$$

$$= 317mV \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Where:-

$V_{LED\ min}$ = 1,2V at 30µA.(Tested) $\qquad$ $V_{bat\ max}$ = 28V.(Spec)

V_{be} = 0,68V $\qquad$ $R1_{max}$ = 120Ω+2% $\qquad$ $R_{X\ max}$ = 1,2MΩ-2%

From the data sheets, TR1 is guaranteed to be off at 50°C with 317mV across the Base/Emitter junction.

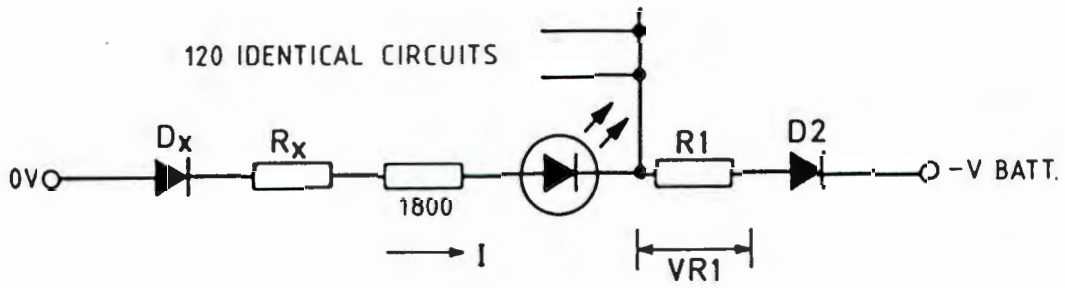


Fig.4.2

The above calculations (including those in the Appendix) have determined:-

a) The values of the Interface Resistors R_X and R_Y
b) The new value of R1 in the Pilot Circuit
c) That the digital Multiplexer/Scanner must be a CMOS device

4.2.2 Sup'y Indicator Interface

The Interface discussed in 4.2.1 above can be connected directly to the Extension and Exchange Indicators. However, due to the constraints of the existing Switchboard Sup'y Indicator design, an extra interface to the Sup'y Indicators is required. This is discussed below:-

a) The voltage allowed across the Sup'y Indicator series current-sense resistor (ref. page 26) should be less than 1V so as to have negligible effect on the Indicator operation. Thus a voltage-gain stage is required to provide correct CMOS logic drive voltages (preferably inverting).

b) The physical construction and location of the Sup'y Indicators in the Switchboard makes it difficult to solder wires to the LEDs as can be done for the other Indicators.

Both the above problems can be solved by unplugging the Sup'y earth sockets, (A.3.1), and plugging them into an Interface Card situated in the control shelf of the Floor Pattern Switchboard. The extra interface takes the form shown in Fig.4.3 below.

Since R_Z and TRZ are referenced to the 0V rail, if the Sup'y Indicator is OFF, then the connection to R_X in the Scanner Interface is effectively open-circuit i.e. the Indicator is sensed as being OFF. If TRZ is ON, then R_X is connected to the 0V rail and the Indicator is sensed as being ON as in the case of the Extension and Exchange Indicators. This result is convenient for the software as all Indicator types have the same logic values for OFF and ON.

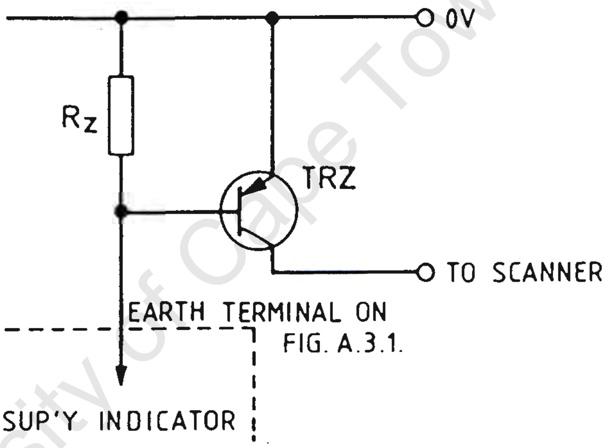


Fig.4.3

NOTE

The SA Relay (for function of Relay see A.3) V_{bat} feed is taken directly from the Switchboard battery or mains PSU only, and is not connected to the 18V standby supply. On mains failure the Sup'y controlled SA Relays, and hence the AWD, will not operate. Thus, the $V_{bat\ min}$ for the Sup'y LED drive current calculation can be taken as 22V, the specified lower limit for the Floor Pattern Switchboard when powered by the normal supply.

The LED drive circuit, (A.3.1), is identical in concept to that of the other Indicators, and can be analysed for $I_{LED\ min}$ as:-

$$I_{LED\ min} = (V_{bat\ min} - V_{sat} - V_{LED\ max}) / 1800$$

$$= 11mA \quad \ldots\ldots \quad (7)$$

Where:-

$V_{bat\ min} = 22V$ $\quad V_{sat} = 0{,}1V$ $\quad V_{LED\ max} = 2{,}2V$

To turn TRZ, (Fig.4.3), on fully using $V_{be\ max}$ = 660mV from (2):-

$$R_{Z\ min} = V_{be\ max} / I_{LED\ min}$$

$$= 60\Omega$$

Take R_Z = 82Ω ±2% as a convenient higher value .... (8)

Also, it is necessary to check the maximum LED current the Sup'y Interface must handle to select a suitably rated transistor for TRZ. Thus:-

$$I_{LED\ max} = (V_{bat\ max} - V_{sat} - V_{LED\ min}) / 1800$$

$$= 15mA$$ .... (9)

If TRZ is chosen to be a BC547B as in the Extension and Exchange Indicator LED drive, then a current-limiting series resistor in the base can be omitted, $I_{b\ max}$ being specified as 200mA for this device.

4.2.3 Summary

a) R_X = 1,2MΩ, ±2% (Indicator Interface)
b) R_Y = 4,7MΩ, ±2% (Indicator Interface)
c) R_Z = 82Ω, ±2% (Sup'y Interface)
d) TRZ is a BC547B (Sup'y Interface)
e) R1 = 120Ω, ±2% (Pilot Circuit. R1 will have to be changed from the existing 120Ω as part of the installation procedure)
f) The Scanner must consist of CMOS devices.

4.3 Scanner and LED Address Decoder

The Scanner and LED addressing circuit should meet the following criteria:-

a) The ability to scan 154 LEDs and transmit the status of each.
b) LS/TTL compatability (for the 8085 microprocessor).
c) Low cost and readily available parts.

Since the Scanner must use CMOS devices, (Chapter 4.2 above), requirement c) is satisfied by using the CD14512 (8 line-to-1 line) multiplexer as the basic digital scanning element. This device is already used by Plessey in other products, thus allowing low cost due to bulk purchases, and ready availability due to stocks currently held for production and spares.

The addressing of 154 LEDs requires 8 control lines ($2^8 = 256$). It was decided to use the the 74LS138 (3 line to 8 line) decoder as the basic element since this is also used by Plessey in current products.

Fig.4.4 overleaf gives the final arrangement and refers in the discussion below:-

a) The scanner consists of 20 CD14512 multiplexers, giving the ability to scan 160 LEDs in total. These are referenced as 1U2, 2U2 up to 20U2. The lower 3 LED address lines, LA0, LA1 and LA2, are common to all multiplexers on inputs A, B, and C. The states of the LEDs on X0 to X7 are connected to the output, Z, according to Table 4.1 below:-

MUX.ADDRESS			INPUTS								O/P
C	B	A	X0	X1	X2	X3	X4	X5	X6	X7	Z
0	0	0	1/0	X	X	X	X	X	X	X	1/0
0	0	1	X	1/0	X	X	X	X	X	X	1/0
0	1	0	X	X	1/0	X	X	X	X	X	1/0
0	1	1	X	X	X	1/0	X	X	X	X	1/0
1	0	0	X	X	X	X	1/0	X	X	X	1/0
1	0	1	X	X	X	X	X	1/0	X	X	1/0
1	1	0	X	X	X	X	X	X	1/0	X	1/0
1	1	1	X	X	X	X	X	X	X	1/0	1/0

Table 4.1

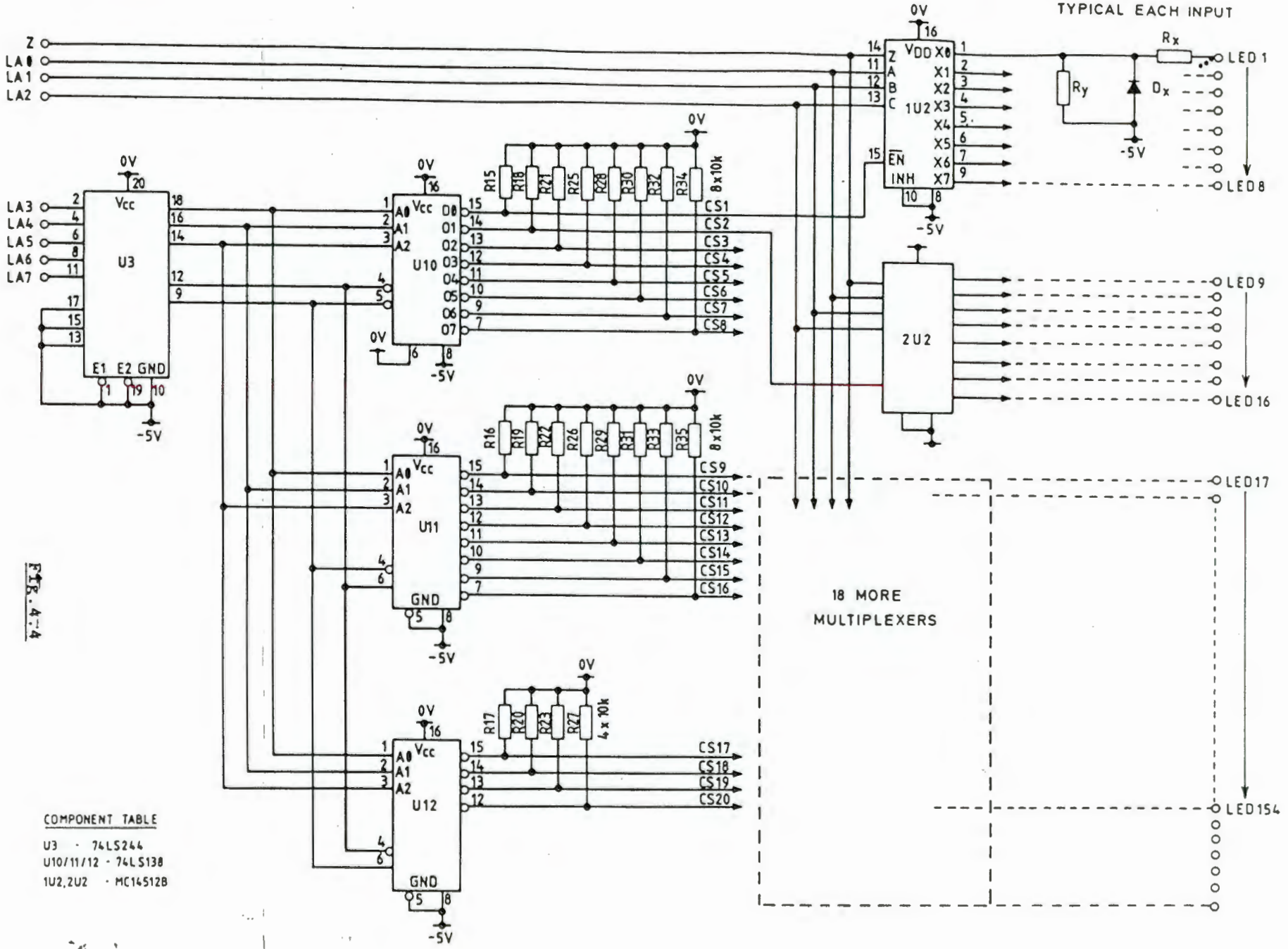

Z
LA0
LA1
LA2
LA3
LA4
LA5
LA6
LA7
U3
E1 E2 GND
U10
U11
U12
GND
Vcc
0V
-5V
R15 R18 R21 R25 R28 R30 R32 R34 8x10k
R16 R19 R22 R26 R29 R31 R33 R35 8x10k
R17 R20 R23 R27 4x10k
CS1
CS2
CS3
CS4
CS5
CS6
CS7
CS8
CS9
CS10
CS11
CS12
CS13
CS14
CS15
CS16
CS17
CS18
CS19
CS20
1U2
2U2
TYPICAL EACH INPUT
Rx
Ry
Dx
LED 1
LED 8
LED 9
LED 16
LED 17
LED 154
18 MORE
MULTIPLEXERS
COMPONENT TABLE
U3 - 74LS244
U10/11/12 - 74LS138
1U2,2U2 - MC14512B

Fig. 4.4

LA7	LA6	74LS138
0	0	U10
0	1	U11
1	0	U12

Table 4.3

From the above it can be seen that the LED addressing circuit works as a "block by block" scheme, summarised below:-

: LA0, LA1, LA2 - Select 1 of 8 LEDs, each multiplexer.
LA3, LA4, LA5 - Select 1 of 8 multiplexers, each 74LS138.
LA6, LA7 - Select 1 of 3 74LS138s.

One final detail is essential. The LS/TTL decoder outputs drive the CMOS multiplexer inputs directly. Unfortunately, as can be seen from Table 4.4, the voltages for a logic "1" are not compatible.

logic	LS/TTL out	CMOS in
0	<0,4V	<1,5V
1	>2,7V	>3,5V

Table 4.4

This problem is overcome by adding "pull-up" resistors to the LS/TTL outputs, referenced as R15 to R35 (Fig.4.4).

The maximum leakage into an LS/TTL output at a logic "1" is specified as 100 µA. The maximum value of pull-up resistor is given by:-

$$R_{p-u\ max} = (V_{cc} - V_{in\ high\ CMOS}) / 10^{-4}$$
$$= 15k\Omega \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$

A resistor network package of 8 10kΩ resistors, with one end of each common, was selected. These networks are extensively used for this purpose in other Plessey products and are thus readily available.

4.4 Clock Generator & Timer Control

The DIGITALKER# SPC (Speech Processor Chip), selected for use in the Blind Operator Aid, requires a 4MHz clock input. There is also a need for a regular, accurately timed interrupt to enable the programme to calculate LED State Timers, Call Timing and a Real Time Clock.

4.4.1 Clock Generator

Fig 4.5 refers in the discussion below:

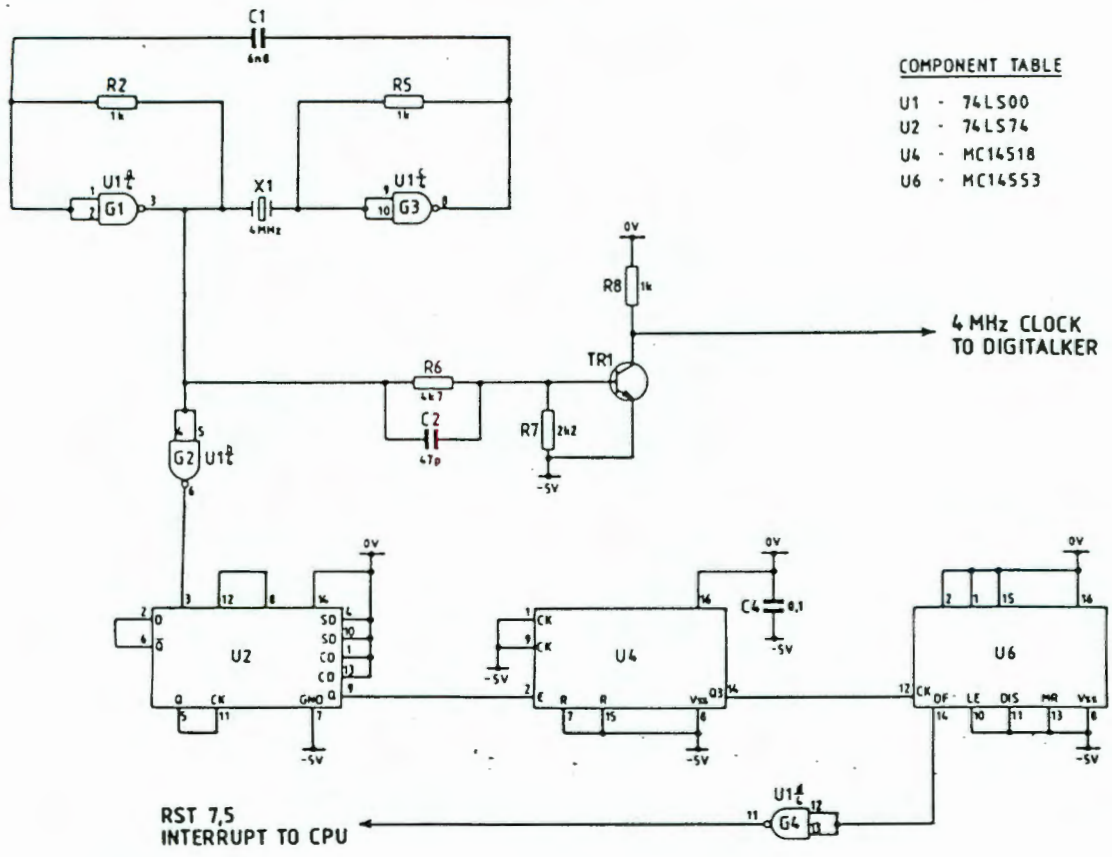


Fig.4.5

The main frequency generator is a 2-gate crystal oscillator.

#DIGITALKER, name registered by National Semiconductor Corporation.

a) The gates G1 and G3 are connected as invertors, biased in the linear region by resistors R2 and R5 connected from output to input. These resistors have a maximum value for LS/TTL of:

$$R_{bias\ max} = (V_{in\ low} - V_{out\ low}) / I_{in\ low}$$

$$= 1600\Omega \qquad (14)$$

1KΩ Resistors are used as a convenient lower standard value.

b) The coupling capacitor, C1, was chosen as 6n8, the impedance at 4 MHz being 6Ω, giving negligible phase shift.

The crystal closes the feedback loop, the whole circuit oscillating at the low-impedance series-resonant frequency of the crystal.

The DIGITALKER data sheet specifies a clock waveform with a minimum 10% to 90% risetime of 25ns. Measurements at point G1/3 (Fig.4.5) on the prototype showed the need for a buffer to provide the specified risetimes. TR1 and associated components perform the buffering function.

a) The collector capacitance of TR1 is 4,5pF maximum. The input capacitance of the DIGITALKER "OSC IN" pin is 5pF maximum, i.e. a total of 9,5pF. Thus R8, the collector pull-up resistor, has a maximum value given by:

$$R8 = -T/(C \ln(1-0{,}9)) \text{ where } T = 25\text{ns},\ C = 9{,}5\text{pF}$$

$$= 1142\Omega \qquad (15)$$

R8 was chosen as 1000Ω ± 2% as a convenient lower standard value.

b) The "speed-up" capacitor C2 in the base drive circuit is necessary to discharge the saturated base storage current of TR1 and provide a rapid switch off on the collector.

Finally, the clock drive to the Timer Control circuit is "squared-up" and buffered by gate G2, connected as an invertor.

4.4.2 Timer Control

Consideration of the three main timing requirements for the software proposed in Chapter 3:-

a) LED cadence timer (0,2s to 3s).
b) Call Timing (up to 30s wait).
c) Real Time Clock (hours, minutes).

showed that a 0,1s interrupt interval was adequate to allow the CPU to achieve a) to c) above without large software overhead due to frequent interrupts. Also, since one 8-bit byte $\equiv$ 256 states, single-byte storage of timers up to 25,6s is possible, leading to simple incrementing and timer routines.

The division ratio from the 4MHz Clock Generator to get a 0,1s signal is thus 400 000, achieved as follows (Fig.4.5):-

a) Divide-by-4: A 74LS74 dual divide-by-2 IC (U2) was chosen as the first divider since the CMOS equivalent has a maximum counting frequency of 3MHz.

b) Divide-by-100: The MC14518, a CMOS dual divide-by-10 IC (U4) was chosen as the middle frequency divider.

c) Divide-by-1000: The MC14553, a CMOS 3-decade IC (U6) provides the final 100ms Timer Control interrupt from the overflow output, buffered by gate G4. The buffered "0"-going signal is 100µs wide.

The Timer Control interrupt is applied to the RST7.5 input of the 8085 CPU, allowing the programme to increment timer bytes.

4.5 Miscellaneous Circuits

These are the Footswitch Debounce, Watchdog, Time-Set switches and PSU.

Fig.4.6 refers in the discussion of 4.5.1, 4.5.2 and 4.5.3:-

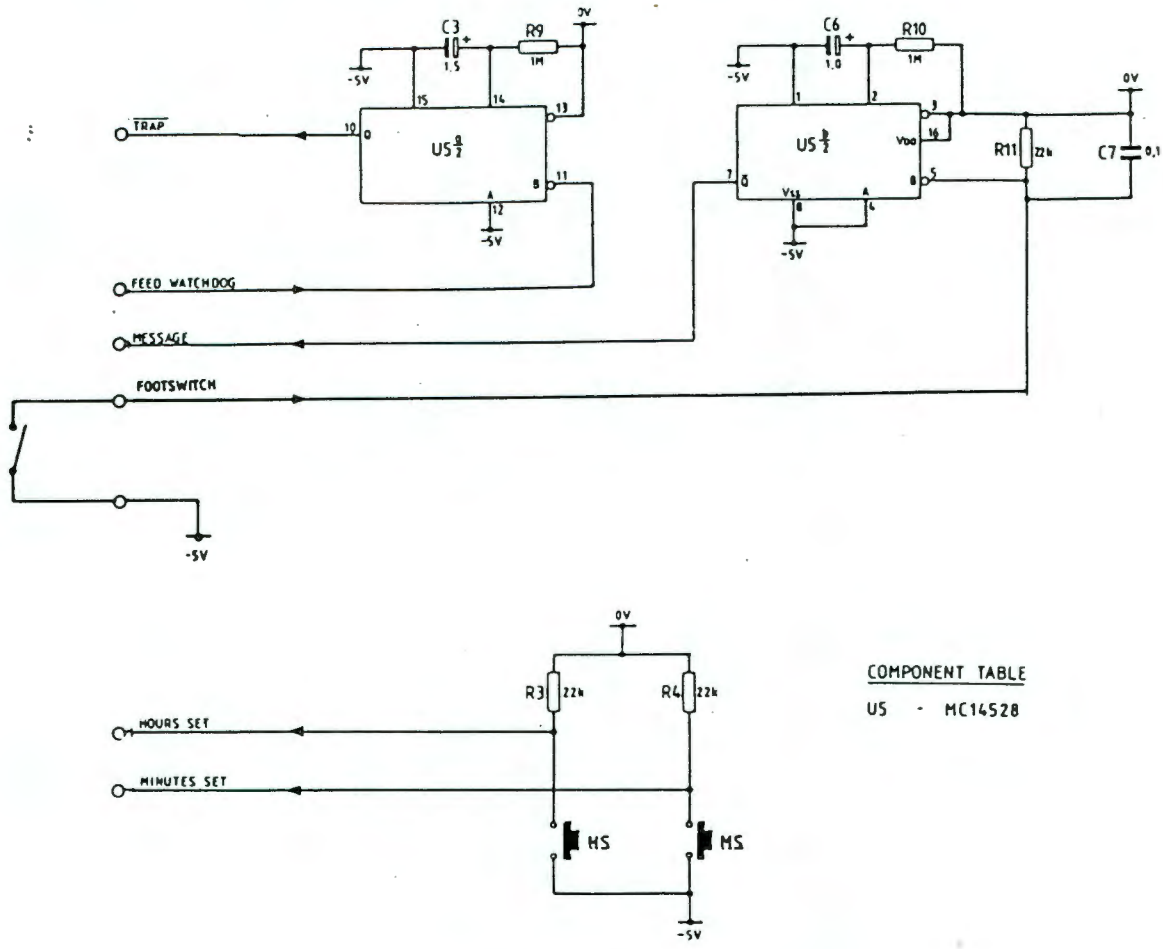


Fig.4.6

4.5.1 Footswitch Debounce

a) The Footswitch, used by the blind operator to signal the Blind Operator Aid to deliver messages, needs to be mechanically robust. A sturdy, inexpensive commercial unit, based on a microswitch in a metal housing was selected. Unfortunately the microswitch contact bounces for up to 20ms.

b) In addition, Footswitch operation time will vary from a short impulse to several seconds (if the operator forgets to lift his/her foot).

Both the above are problems solved by using one half of an MC14538 dual CMOS IC multivibrator, U5b, connected in the non-retriggerable mode. This configuration will ignore contact bounce less than the triggered period and provide one output pulse per Footswitch operation (to ignore permanent closure).

1) All of the messages required from the Blind Operator Aid last longer than 0,5s. A time of 0,3s, set by C6/R10 and U5b, seemed adequate.

2) The MC14538 is triggered only on a change of voltage from a logic "1" to a logic "0" on pin 5 and will thus ignore a permanently operated Footswitch.

4.5.2 Watchdog

The Watchdog sanity control can take several forms, two of which are described below:-

a) An extra divider chain driven from the 100ms interrupt. The output from this extra chain can control the Master Reset, so that if the 100ms interrupt is ignored then the CPU is forced to restart the programme.

b) A monostable one-shot that is kept in the triggered state by the normal programme execution. If the programme flow is corrupted and the one-shot times out, then the change of output can interrupt the CPU and cause a programme restart.

b) Was selected as the cheapest and most flexible method since the Footswitch Debounce circuit leaves spare one half of an

MC14538 and the timeout is easily adjustable (by C3/R9). The initial timeout was set as 1,5s for the field trials.

The one-shot trigger on U5a pin 5 responds only to a change of level from a logic "1" to a logic "0". The CPU must therefore write several sequential defined bytes to the I/O lines, ensuring that a regular pass through the relevant programme section is needed to retrigger the Watchdog one-shot.

4.5.3 Set Time Switches

These switches allow the operator to set both the Hours and Minutes of the software-driven Real Time Clock. They are arranged to put a logic "0" on two of the I/O lines, pull-up resistors R3 and R4 providing a logic "1" when the switches are not operated.

Debounce can be provided by the programme by looking for two sequential "0" levels at 100ms intervals.

4.6 Power Supply Unit

A commercial 220V mains-driven unit, suitable for mounting on a PCB, is used to power the Blind Operator Aid. The 12V/1A DC output is isolated electrically from the mains input earth, allowing the provision of both negative and positive supply voltages as required for the Scanner/Decoder and DIGITALKER. The 9V supply for the speech power-amplifier and the 5V supply for the CPU and Scanner/Decoder were derived from the 12V supply using standard IC regulators.

The circuit diagrams for the supplies are included in the Appendix for reference only.

4.7 Speech Synthesizer

Several different types of Speech Synthesizer were considered for the Blind Operator Aid. These are discussed briefly below:-

a) PCM

For a more detailed discussion on PCM (Pulse Code Modulation) speech transmission and encoding, the reader is referred to the following texts as given in the Bibliography:-

Bib. 5) National Semiconductor Corp. "Audio Data Book"
Bib.11) CCITT Recommendations G732 & G711 for PCM
Bib.12) Ciarcia on "ADPCM for Highly Intelligible Speech"
Bib.13) Bennet on "PCM and Digital Transmission

PCM speech storage and replay gives telephone quality speech with a frequency range of 300 to 3000Hz.

To store speech, the waveform is sampled every 125μs i.e. at an 8kHz rate. The voltage samples are encoded by an ADC (Analogue to Digital Converter) and the resultant binary numbers stored in sequential locations in a ROM (Read Only Memory). On replay, the stored numbers are converted to analogue voltages by a DAC (Digital to Analogue Convertor), each voltage value being held for 125μs. The stepped output waveform is filtered to remove the abrupt transitions and restore the original speech quality.

A 12-bit binary word (Bib.13) is required to give good speech reproduction if the samples are encoded in a linear manner i.e each sequential binary value increment is equal to the same analogue voltage increment. The 12-bit numbers, however, produce a large amount of data to be stored in ROM.

Fortunately, the human ear becomes progressively less sensitive to errors in the conversion process as the speech levels increase, allowing amplitude compression of the encoding and decoding at the higher levels i.e. successive equal binary number increments are proportional to larger and larger voltage increments. This compression is usually carried out according to one of two laws as specified by the CCITT, A-law (used in South Africa and Europe) or μ-law (used in the U.S.A. and Canada) (Bib.11).

This compression allows good speech reproduction using a series of 8-bit numbers (Bib.11 & 13). The MSB (Most Significant Bit) denotes the sign of the voltage, leaving 7 bits to represent ±127 voltage levels. Commercial IC CODECS (COder DECoders) and Filters using an 8-bit compression law are used in modern digital Private Telephone Exchanges such as the Plessey K2# Digital PEBX.

DPCM (Differential PCM) is a variant of ordinary PCM where only the differences between successive binary numbers are stored, these differences being added each time to get the new value (Bib.5 & 12).

ADPCM (Adaptive DPCM) varies the step size of each of the difference values to cope with rapidly varying voltages. DPCM gives a 25 to 50% reduction in data storage needs (Bib.12).

The words etc. must be encoded if standard off-the-shelf Codecs and Filters are used. This requires the development of the encoding control circuitry as well as the time taken to edit the resultant speech.

There are 40 basic speech elements (letters, numerals and words) required from the Blind Operator Aid. (See DIGITALKER later).

K2 is a trademark of Plessey South Africa Limited

At an average length of 0,5s for each element, this will require 20s of stored speech, plus the overhead of stop, start and other control bytes. At the 8kHz sample rate, this requires 160 kbytes of ROM, more than can be directly controlled by an 8-bit microprocessor using 16 address lines.

OKI, a Japanese company, supply a kit of ICs at a cost of R150 to perform the encoding and decoding functions using ADPCM to get the equivalent speech quality as 12-bit processing (Bib.9). The desired 20s of speech needs 48 kbytes of 8-bit wide ROM. The start address of a speech element is loaded into presettable address counters, which are then automatically incremented until the end of the desired word is reached.

Since the object of this investigation was not Speech Synthesis per se, neither the cost of the circuitry nor the time needed to develop the vocabulary was considered justified.

b) Phoneme Addition Devices such as the VOTRAX module produce speech by the sequential addition of basic speech elements, called phonemes, such as the consonants, vowels, voiced and unvoiced fricatives. The vowel phonemes produced by the module can be lengthened to change the sound of the word.

Typical phrases using the Votrax module were synthesized and recorded on tape.

Listening tests on the resultant speech were done by both the author and the SANCB. The subjective feeling was that the metallic and rather nasal quality of the speech would lead to operator fatigue and irritation.

e) The pitch of the stored waveforms in c) above can be varied by altering the rate of playback of the data stored for any section, adding inflection to the speech.

f) The amplitude of the stored waveforms in c) above can be varied by altering the gain of the playback amplifier, adding emphasis to the speech.

Since the data storage is greatly reduced using the criteria a) to f) above, the actual encoding and decoding of the waveforms in c) can be performed using ADPCM (Bib.12) to provide accurate reproduction of the voiced sounds.

The output waveform needs low-pass filtering on playback to remove the high frequency noise introduced by the ADPCM encoding and decoding.

Drawbacks of the above method of data compression are that the process can only be performed using expensive and complex equipment, including a powerful minicomputer, and that the playback circuitry is too complex to be realised with even medium scale integrated circuits.

Fortunately, commercial LSI (Large Scale Integration) modules such as the National Semiconductor DIGITALKER are readily available, complete with ROMs, to provide letters, numerals, tones, silence periods and a range of words suitable for use with measuring instruments and control functions (and fortunately the Blind Operator Aid).

Table 4.5 overleaf gives the speech output list and corresponding 8-bit addresses to be supplied by the CPU to the DIGITALKER.

Word	8-Bit Binary Address SW8 SW1
THIS IS DIGITALKER	00000000
ONE	00000001
TWO	00000010
THREE	00000011
FOUR	00000100
FIVE	00000101
SIX	00000110
SEVEN	00000111
EIGHT	00001000
NINE	00001001
TEN	00001010
ELEVEN	00001011
TWELVE	00001100
THIRTEEN	00001101
FOURTEEN	00001110
FIFTEEN	00001111
SIXTEEN	00010000
SEVENTEEN	00010001
EIGHTEEN	00010010
NINETEEN	00010011
TWENTY	00010100
THIRTY	00010101
FORTY	00010110
FIFTY	00010111
SIXTY	00011000
SEVENTY	00011001
EIGHTY	00011010
NINETY	00011011
HUNDRED	00011100
THOUSAND	00011101
MILLION	00011110
ZERO	00011111
A	00100000
B	00100001
C	00100010
D	00100011
E	00100100
F	00100101
G	00100110
H	00100111
I	00101000
J	00101001
K	00101010
L	00101011
M	00101100
N	00101101
O	00101110
P	00101111

	8-Bit Binary Address SW8 SW1
Q	00110000
R	00110001
S	00110010
T	00110011
U	W0110100
V	00110101
W	00110110
X	00110111
Y	00111000
Z	00111001
AGAIN	00111010
AMPERE	00111011
AND	00111100
AT	00111101
CANCEL	00111110
CASE	00111111
CENT	01000000
400HERTZ TONE	01000001
80HERTZ TONE	01000010
20MS SILENCE	01000011
40MS SILENCE	01000100
80MS SILENCE	01000101
160MS SILENCE	01000110
320MS SILENCE	01000111
CENTI	01001000
CHECK	01001001
COMMA	01001010
CONTROL	01001011
DANGER	01001100
DEGREE	01001101
DOLLAR	01001110
DOWN	01001111
EQUAL	01010000
ERROR	01010001
FEET	01010010
FLOW	01010011
FUEL	01010100
GALLON	01010101
GO	01010110
GRAM	01010111
GREAT	01011000
GREATER	01011001
HAVE	01011010
HIGH	01011011
HIGHER	01011100
HOUR	01011101
IN	01011110
INCHES	01011111

	8-Bit Binary Address SW8 SW1
IS	01100000
IT	01100001
KILO	01100010
LEFT	01100011
LESS	01100100
LESSER	01100101
LIMIT	01100110
LOW	01100111
LOWER	01101000
MARK	01101001
METER	01101010
MILE	01101011
MILLI	01101100
MINUS	01101101
MINUTE	01101110
NEAR	01101111
NUMBER	01110000
OF	01110001
OFF	01110010
ON	01110011
OUT	01110100
OVER	01110101
PARENTHESIS	01110110
PERCENT	01110111
PLEASE	01111000
PLUS	01111001
POINT	01111010
POUND	01111011
PULSES	01111100
RATE	01111101
RE	01111110
READY	01111111
RIGHT	10000000
SS (Note 1)	10000001
SECOND	10000010
SET	10000011
SPACE	10000100
SPEED	10000101
STAR	10000110
START	10000111
STOP	10001000
THAN	10001001
THE	10001010
TIME	10001011
TRY	10001100
UP	10001101
VOLT	10001110
WEIGHT	10001111

Table 4.5

The speech elements needed by the Blind Operator Aid to provide supervision of all call types are available from the list as shown below:-

a) Digits 1 to 19.

b) Multiples of ten i.e. 20 to 100.

c) Alpha characters i.e. X (for Extension), L (for Loop).

d) Silent periods between words.

e) Tones to signal events such as Ready-to-start.

f) Words such as Cancel, Wait (Weight), Set, The, Time. Is, Hour, Minute, Ss (for plurals), In (for INcoming), and Danger (to signal a malfunction).

Fig.4.7 overleaf refers in the discussion below.

The SPC (Speech Processor Chip), U7, interfaces directly with the microprocessor data and control lines, words being accessed by an 8-bit Word Select Address (on the CPU data bus) and a WRITE signal. The CS (Chip Select) on U7 is connected to one of the CPU I/O lines, being driven to a logic "0" only when addressing the SPC. This arrrangement removes the need for an extra decoder.

The SPC accesses the Speech ROMs via 14 address lines, A0 to A13 (A13 is used as the ROM CS), and 8 data lines.

The INTR interrupt line from U7 is used to signal the CPU via one of the I/O lines that the DIGITALKER is outputting speech, remaining at a logic "1" while busy.

The OSC IN on U7 is driven by the 4MHz buffered clock with 25ns maximum rise and fall times (4.4.1).

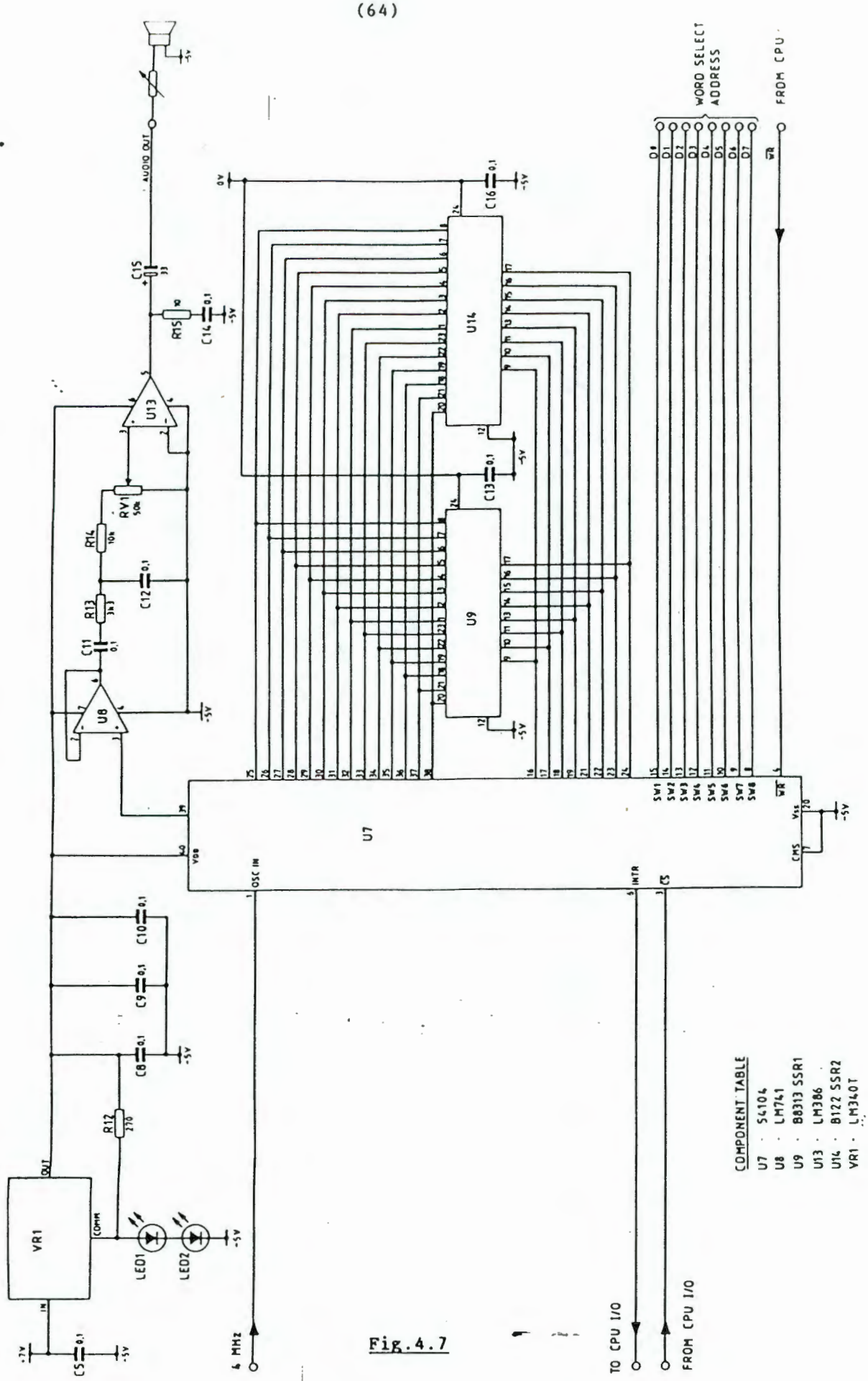

AUDIO OUT
C15
R15
C14
U13
RV1
R14
C12
R13
C11
U8
U14
U9
C16
C13
U7
OSC IN
VDD
INTA
CS
CMS
VSS
WORD SELECT ADDRESS
FROM CPU
SW1
SW2
SW3
SW4
SW5
SW6
SW7
SW8
C10
C9
C8
R12
VR1
OUT
COMM
IN
LED1
LED2
C5
4 MHz
TO CPU I/O
FROM CPU I/O
COMPONENT TABLE
U7 - S4104
U8 - LM741
U9 - B8313 SSR1
U13 - LM386
U14 - B122 SSR2
VR1 - LM340T

Fig.4.7

The speech voltage output on U7 pin 39 is buffered by amplifier U8, filtered by R13/C12 (discussed further after Fig.4.9) and then amplified by the audio- power amplifier U13 to provide 250mW maximum power output to the loudspeaker, via C15. RV1 is the volume control.

VR1, the SPC V_{dd} and audio-power amplifier supply regulator, is fed from the +7V rail. VR1 is referenced to -5V is by means of two LEDs to give an output of +9V w.r.t. the -5V supply. Decoupling to high frequencies to prevent possible feedback instability due to long leads is provided by C5/C8/C9/C10.

To remove any confusion in the mind of the reader, Fig.4.8 shows the multiple power-referencing of the entire Blind Operator Aid, necessitated by the restraints of interfacing to the Floor Pattern Switchboard.

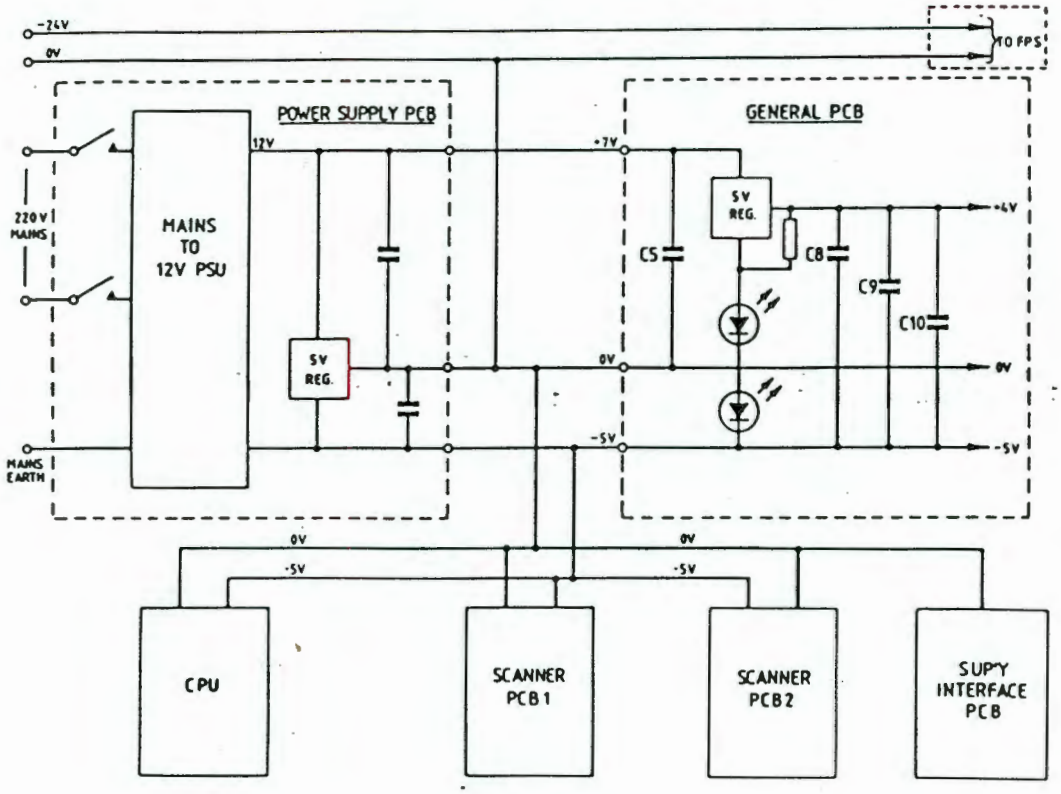


Fig.4.8

Filtering NS (National Semiconductor) apply a high frequency pre-emphasis to the recorded speech before analysis. The sampling and ADPCM introduce further high frequency noise components. A low-pass filter with a roll-off at 200Hz to compensate for the pre-emphasis and to attenuate the noise is recommended by NS. Additionally, a high-pass roll-off at 200Hz (optional, dependant upon louspeaker response) to attenuate low frequency noise is also suggested. The low-pass filtering is provided by R13/C12, the high-pass by R13/C11 (Fig.4.8).

4.8 Overall Circuit

The reference numbers used for the ICs, resistors, capacitors and other components in Sections 4.2 through 4.7 above have purposely been kept the same as those on the individual circuit diagrams in the Appendix. Thus, the reader will find that the detailed functions of these diagrams may be found in the sections. A brief guide to the placement of the individual circuits is given below:-

a) Scanner PCB There are two of these, allowing scanning of 160 LEDs. Each PCB contains 10 CD14512 CMOS multiplexers, the pull-up and pull-down resistors (R_X and R_Y), the protection diodes D_X and the sockets for the ribbon cables to the Indicator LEDs.

b) Sup'y Interface PCB This PCB is housed in the shelf of the Floor Pattern Switchboard, and contains the 34 interface inverter transistors (TRZ in text), the 82Ω series resistors (R_Z in text), the plugs that connect to the Sup'y Indicators and the sockets for the Scanner ribbon cables.

c) General PCB This contains the Footswitch Debounce, Watchdog, Set Time switches, Scanner Address Decoder, DIGITALKER, audio-power amplifier and 9V Regulator.

d) Power Supply PCB This provides a mounting means for the +12V PSU and the -5V Regulator.

e) CPU PCB This is the standard module suggested by SAPO. The circuit diagram is included for reference only.

4.9 Construction Notes

The loudspeaker and volume control are mounted on the front panel and are the only components not mounted on a PCB module.

The PCBs are mounted on sections of extruded aluminium panel and held in the Blind Operator Aid case by two knurled finger-screws. Thus:-

a) The Mains-switch and the socket for the Mains Cord are mounted on the PSU PCB panel.

b) The Set Time switches and the socket for the Footswitch are mounted on the General PCB panel.

c) The ribbon cables to the Indicator LEDs are taken out through cutouts in the Scanner PCB panel. 16 Ribbon cables, each 3m long, are supplied with each unit.

The PCB connections are taken out through DIN-41612 64-way plugs mounted on the inboard end of the PCB. The interconnections are presently made by wire-wrapping the pins of mating DIN-type sockets to allow changes after the field trials are completed, at which stage a PCB-type motherboard will be designed and fitted.

5.1 General

The software development for the Blind Operator Aid served a two-fold purpose:-

a) To provide software control for the Blind Operator Aid.
b) To serve as a training aid for teaching structured programming in a working environment.

The logic of the main programme and sub-routines was laid out using P-Code (see below) and then hand-compiled to produce the Assembler Source Code. The Assembler Code was then processed in the normal manner to produce the machine Object Code.

5.2 P-Code

P-Code (Pseudo Code) is a flexible method of developing programmes using Constructs (Logical Structures) and Data Structures gleaned from BASIC, PASCAL, C or any other useful HLL (High Level Language). Since there is no "official" definition of P-Code, it is possible for the programmer to use Constructs such as IF-THEN-ELSE (from BASIC and PASCAL), Data Structures such as Records, Arrays and Vectors (taken from C) and to include efficient Assembler Code (referred to as Embedded Assembler) for critical time-dependent processes.

During the early work on the Blind Operator Aid, the P-Code was compiled to run on the MC6800 8-bit microprocessor and tested using a 6800-based prototype which incorporated the DIGITALKER. The "debugged" P-Code was subsequently re-compiled to run on the 8085-based CPU module. This resulted in the Object Code being increased by 15%, mainly due to the 8085 single Accumulator and lack of Index-offset addressing.

To digress, since no microprocessor-dependent compiler is used, the P-Code can be converted for any target processor, allowing demonstration of the differences between the Assembler Codes that are produced. For example, the effect of the different

microprocessor architectures on the Assembler Code. In this case, the 6800/6809 from Motorola can do arithmetic in 2 Accumulators, A & B, whereas the 8085 from Intel has only one Accumulator, A. However, the 8085 has more Index Registers than the 6800, and so on.

5.2.1 Constructs

The Logical Constructs used in the P-Code are discussed below, preceded by some definitions:-

(statements) Refers to actions such as (A = B + C). Note that (A = B + C) does not have the same meaning as the Algebraic Equality, A = B + C, where the L.H.S. is already equal to the R.H.S. Rather, it means the L.H.S. is now to be given the value of the R.H.S. In fact, in some versions of BASIC, the actual statement is :-

```
LET A = B + C;
```

This concept is called "Assignment" and makes sense of the statement:-

```
(A = A + 1);
```

(predicate) Refers to a proposition such as (A = B), as in:-

```
IF A = B THEN etc.; (overleaf).
```

(variable) Refers to a value that can be altered by statements in the programme e.g. operations on the variable "TOTAL" could alter the contents of a byte in RAM, the address of the byte being known to the programme by the Variable Name, TOTAL.

(const) Refers to a constant i.e. a value that does not change, irrespective of the statements in the programme e.g. the constant WEEKDAYS could be assigned the value 5, whereas WEEKEND would be assigned the value 2.

```
d) REPEAT UNTIL (predicate)
      BEGIN
         (statements)
      ENDREPEAT;
```

In contrast to c) above, this looping Construct tests the condition after the pass through the statements. Thus, the statements are always executed at least once.

```
e) CASE SELECT OF (variable)
      BEGIN
         CASE(value1)
            BEGIN
               (statements)
            ENDCASE(value1);
         ↓
         CASE(valuen)
            BEGIN
               (statements)
            ENDCASE(valuen);
      ENDCASE;
```

This Construct executes only one set of statements dependent upon the (value) of (variable). It is equivalent to but easier to follow than the PL/1 Construct:-

```
   IF (predicate) THEN

   ELSE IF (predicate) THEN
         ↓
   ELSE IF (predicate) THEN;
```

```
f) LINK (process name)
      BEGIN
         (assembler code)
      ENDLINK;
```

This construct enables Embedded Assembler to be included.

NOTE

a) For controlled programme flow, there are no GOTO or ELSE GOTO constructs which allow the programme to jump out of loops or statements at random. These jumps are reserved in P-Code for GOTO ERROR or ELSE GOTO ERROR only. This forces the programmer to ensure that all PROCEDURES (sub-routines) have a consistent structure i.e. there is only one entry point and only one exit point.

The author has worked in the past on an electronic PEBX with 24Kbytes of ROM written in Assembler Code where the above rule was ignored. Modification of the code to provide extra features etc. was extremely time-consuming due to the need for finding and checking the effects of changes on all the scattered entries and exits of sub-routines.

b) The "BEGIN...END" feature breaks the statements into blocks, ensuring that there is no confusion as to which statements belong to which construct.

5.2.2 Data Structures

The Data Structures used in the Blind Operator Aid P-Code are:-

a) CONST:

(constx) = (valuex);

where (value) can be Decimal (XXD), Binary (XXXXXXXXB) or Hexadecimal (0XXXXH).

b) VAR:

```
(var1) = BYTE;                          (8 bits wide)
(var2) = WORD;                          (16 bits wide)
(var3) = RECORD OF BITS (a to h)
            BITa = (vara);
                    ↓
            BITh = (varh);
(var4) = RECORD OF BYTES (x to z)
            BYTEx = (varx);
                    ↓
            BYTEz = (varz);
```

c) TYPE:

```
(var) = (type);
```

e.g. LAMPS = RED, BLUE, GREEN;

5.2.3 Procedures

All Procedures (sub-routines that perform simple tasks) have the form:-

```
PROCEDURE "name";
  BEGIN
    (statements and constructs)
  END;
RETURN;
```

Within a Procedure, successive indenting of lower levels of constructs (and data definitions) leads to ease of reading. The use of a ";" or similar sign after a statement, construct or data definition acts as a separator, being used by a software-driven machine compiler to indicate individual actions to be performed. Strict use of it helps the programmer to produce structured modules.

Listings of the Procedures for the Blind Operator Aid using the above principles are given in the Appendix.

a) Full Name: MAIN Abbreviation: MAIN

P-Code Statement: None, is entered after CPU Reset.

Procedures Called: TIMER, TIMESET, LAMPSCAN, LAMPSTATE, LOOPSTATE, SORT-EXTENSION, SORT-EXCHANGE, SORT-LOOPS, OUTVOICE, GENERATE-MESSAGE

Purpose:

This is the main Procedure that polls all other Procedures to perform the Blind Operator Aid tasks. It initialises the variables in RAM, sets up the control of the I/O hardware, checks the memory for faults and then loops forever, performing the Blind Operator Aid programme.

Main Tasks:

a) Sets software stack-pointer.

b) Tests the RAM for hardware faults.

c) Checks the ROM for hardware faults.

d) If the hardware passes, gives the sign-on tones via the DIGITALKER, hence exercising the speech synthesizer.

e) Initialises Message Stack Pointers, Message control variables and the initial state of the LEDs as OFF.

f) Thereafter performs the Blind Operator Aid tasks such as scanning the LEDs, updating timers, generating messages and controlling the DIGITALKER to respond to the operator depressing the Footswitch.

g) Provides the Interrupt Service Routine (it sets an Interrupt Flag). At the end of each set of the normal tasks the software goes into a loop, checking for a "Set" Interrupt Flag. When detected, the Flag is reset and a new cycle of tasks is started.

Notes:

a) The ROM is checked by summing all bytes except the last and comparing the result to a Checksum stored in the last byte.

b) The polling of all other tasks is synchronised to the 100ms interrupt as discussed in g) above.

c) The Watchdog is reset after each 100ms interrupt.

b) Full Name: TIMER Abbreviation: TIMER

P-Code Statement: "GO UPDATE CLOCK TIMER"(,,,,);

Procedures Called: None

Purpose:

This Procedure is called at the start of every 100ms period. Tenths of a second, Seconds, Minutes and Hours are updated.

Main Tasks:

a) Update the Real Time Clock.

c) Full Name: TIMESET Abbreviation: TIMSET

P-Code Statement: "GO SET REALTIME CLOCK"(,,,,),(,,,,);

Procedures Called: BINARY-TO-BCD, TRANSFER-MESSAGE

Purpose:

This Procedure allows the operator to set the Real Time Clock to the nearest minute by means of the SET HOURS and SET MINUTES switches.

Main Tasks:

a) Debounces the switches.
b) Generates either of the messages "SET HOURS" or "SET MINUTES" as appropriate for the DIGITALKER to speak.
c) Increments Hours or Minutes and announces via DIGITALKER the result.
d) Performs c) above every 1,5s to allow the operator time to react and release the switch when needed.

Notes:

a) If both switches are operated simultaneously, they are ignored.

d) Full Name: LAMPSCAN Abbreviation: LMPSCN

P-Code Statement: "GO SCAN ALL THE LAMPS"(,,,,);

Procedures Called: None

Purpose:

This Procedure scans all the LEDs and puts the state (1 ≡ ON, 0 ≡ OFF) in the LSB of sequential bytes in the RAM, referred to as LAMPSTACK, ready for the Procedure LAMPSTATE to process.

e) Full Name: LAMPSTATE Abbreviation: LMPSTT

P-Code Statement: "GO CHECK LAMP STATUS"(,,,,);

Procedures Called: None

Purpose:

This Procedure keeps track of the sequential states of each LED (read every 100ms), thus deciding whether the LED is actually ON or OFF, or starting to be timed as ON or OFF (LED debounce to eliminate spurious noise states). The bits representing ON or OFF in the byte associated with each LED are set or reset accordingly.

f) Full Name: LOOPSTATE Abbreviation: LOPSTT

P-Code Statement: "GO CHECK LOOP STATUS"(,,,,);

Procedures Called: None

Purpose:

Since the meaning of a given combination of Sup'y LEDs depends upon the states before the latest change, this Procedure provides a one-step memory. The two bits representing the states of an LED pair at each reading are placed to the right of the bits for the previous reading in the byte associated with each LED pair, the bits for any other previous readings being discarded. The four bits thus obtained and stored are used by Procedure SORT-LOOP to point to the next process.

g) Full Name: SORT-EXTENSION Abbreviation: SRTEXT

P-Code Statement: "GO SORT EXT. MESSAGES"(,,,,);

Procedures Called: EXTENSION-MESSAGE

Purpose:

This Procedure maintains a list of which Extensions are calling the operator by checking through the bytes associated with the Extension LEDs, putting the number of the LED onto the top of the Extension Message Stack if the LED is ON and the number is not already on the stack. Numbers are placed on the stack in a First-in-First-out manner. If the LED goes OFF, then the number is removed from the stack and any numbers above are moved downwards to close the gap. The number of messages in the stack, stored in a byte at the bottom of the stack, is also updated.

h) Full Name: SORT-EXCHANGE Abbreviation: SRTEXC

P-Code Statement: "GO SORT EXC. MESSAGES"(,,,,);

Procedures Called: EXCHANGE-MESSAGE

Purpose:

This Procedure is identical to SORT-EXTENSION above except that it deals with the Exchange LEDs and the Exchange Message stack.

i) Full Name: SORT-LOOP Abbreviation: SRTLOP

P-Code Statement: "GO SORT LOOP MESSAGES(,,,,);

Procedures Called: STATES

Purpose:

This Procedure checks through the bytes associated with the Sup'y LED pairs and finds a State Name corresponding to the four bits produced by LOOPSTATE. The State Name is found in a look-up table called STATE-TABLE, the State Name being passed as a parameter to STATES for processing. This produces a very flexible hardware unit, as the response to a State is easily altered by changing the STATE-TABLE entries.

j) Full Name: STATES Abbreviation: STATES

P-Code Statement:

"DO PROCEDURE ->POINTER"(,LOPNUM,,STAPTR,);

Procedures Called: LOOP-MESSAGE

Purpose:

This Procedure is actually a set of sub-Procedures, one for each of the State Names in STATE-TABLE in Procedure SORT-LOOP. The parameter LOPNUM (the Cord Circuit number) is for use by Procedure LOOP-MESSAGE when placing messages on the Loop Message Stack. The parameter STAPTR (the pointer to the Cord Circuit information bytes) is used by the sub-Procedures when updating call information such as Waiting-time, number of telephone switch-hook depressions (when calling the operator during a call), OFF/ON Timers and status markers such as Loop-ON or Loop-OFF.

m) Full Name: LOOP-MESSAGE Abbreviation: LOPMES

P-Code Statement:

```
"DO LOOP MESSAGE"(OFF,LOPNUM,,,);
"DO LOOP MESSAGE"(ON,LOPNUM,CANCEL,,);
"DO LOOP MESSAGE"(OFF,LOPNUM,CHKLOP,,);
```

Procedures Called: None

Purpose:

This Procedure, called by STATES, does the actual insertion (parameter ON) and deletion (parameter OFF) of the Loop Number (parameter LOPNUM) on the Loop Message Stack. If the parameter ON is passed, then the extra parameters CANCEL and CHKLOP, added to the number on the stack, tell GENERATE-MESSAGE to produce "CANCEL LOOP" and "LOOP WAITING" messages respectively from the DIGITALKER when the Footswitch is operated.

n) Full Name: GENERATE-MESSAGE Abbreviation: GENMSG

P-Code Statement: "GO GENERATE MESSAGES"(,,,,);

Procedures Called: GENERATE-WORDS, EXTENSION-MESSAGE, EXCHANGE-MESSAGE, LOOP-MESSAGE

Purpose:

This Procedure generates the sequential bytes that control the DIGITALKER in response to the Footswitch operation. The priority of messages is:-

a) All Loop Messages.

b) All Exchange Messages.

c) All Extension Messages

d) If none of the above exist, the Time is given.

The messages b) and c) are repeated twice if the LED remains ON, then being placed at the back of the relevant Message Stack to give other Messages a chance. Messages are removed from the stack if the LED goes OFF, indicating a succesfully answered call. "CANCEL LOOP" messages are produced once only. "LOOP WAITING" messages are repeated until the Sup'y LEDs change to another state, indicating operator action.

o) Full Name: GENERATE-WORDS Abbreviation: GENWRD

P-Code Statement:

```
"GO GENERATE WORDS"(CANCEL,LOPNUM,,,);
"GO GENERATE WORDS"(CHKLOP,LOPNUM,,,);
"GO GENERATE WORDS"(EXTENS,EXTNUM,,,);
"GO GENERATE WORDS"(EXCHNG,EXCNUM,,,);
"GO GENERATE WORDS"(TIME,,,,);
```

Procedures Called: TRANSFER-MESSAGE, BINARY-TO-DATA

Purpose:

This Procedure is used by GENERATE-MESSAGE to produce the sequential bytes that control the DIGITALKER when synthesizing the voice output. The message formats are stored in a table called MESSAGES. The bytes are placed in a Word Buffer in RAM, ready for the CPU to write to the DIGITALKER. The actual transfer from MESSAGES to the Word Buffer is performed by TRANSFER-MESSAGE. The parameters CANCEL, CHKLOP, EXTENS, EXCHNG and TIME indicate which type of message is to be generated.

p) Full Name: TRANSFER-MESSAGE Abbreviation: TFRMSG

P-Code Statement:

```
"GO TRANSFER MESSAGE"(,,MSGPTR,,);
```

Procedures Called: None

Purpose:

This Procedure transfers the message, controlled by the format bytes, from the table in MESSAGES to the output Word Buffer in RAM. The parameter MSGPTR is used to point to the particular message in the table. The message format indicates whether a fixed word, e.g. "CANCEL", or a variable, e.g. an LED number, is to be transferred.

q) Full Name: OUTVOICE Abbreviation: OUTVCE

P-Code Statement: "GO PUT SPEECH MESSAGE"(,,,,);

Procedures Called: None

Purpose:

This Procedure takes the bytes in sequence from the output Word Buffer and writes them to DIGITALKER one at a time until the Buffer is empty. A check is made to see if DIGITALKER is ready for the next word (by reading the synthesizer Ready line via the CPU I/O device).

NOTE This Procedure is called every 100ms, synchronised to the Interrupt as discussed in Procedure MAIN. This results in a possible extra 100ms silence period being inserted between the words of a message due to DIGITALKER not being driven. During listening tests this random extra silence was not easily noticable.

r) Full Name: MESSAGES Abbreviation: MESSGS

P-Code Statement: None

Procedures Called: None

Purpose:

This Procedure is really only a table of Message Names and format bytes. The flexibility in producing messages results from the ease with which bytes can be redefined without affecting any other Procedure.

5.3.2 Software Comments

There are several comments relevant to the Procedures summarised in 5.3.1:-

a) The use of a Word Buffer allows the Blind Operator Aid programme to generate the information for the DIGITALKER, then output the first word and continue with the routine tasks of scanning the LEDs, updating the Real Time Clock etc., without first waiting for the complete message to be spoken. The next word in any sequence is produced only if the DIGITALKER is ready and the programme is busy with a new 100ms task cycle. The speech element length of the message is thus unimportant.

b) The use of look-up tables for the Voice Output response to the LED states, Set Time switches and Footswitch in Procedure MSSGS allows the ergonomics of the Blind Operator Aid to be changed with relative ease.

The messages chosen for the field trial units are given below:-

a) Extension LED ON Designated as "X" followed by number.

1) e.g. "X TWELVE" for numbers up to and including 20.

2) e.g. "X FIVE SIX" for numbers greater than 20, but not multiples of 10.

3) e.g. "X FIFTY" For numbers that are multiples of 10.

b) Exchange LED ON Designated as "IN" followed by number.

1) e.g. "IN SIX" for numbers up to and including 20.

c) Sup'y Indicator Designated as "L" followed by number and instruction.

1) e.g. "CANCEL L TWELVE" for numbers up to and including 17 for any Cord Cicuits to be disconnected.

2) e.g. "L FIVE WAIT" for numbers up to and including 17 for any Cord Circuits where:-

a) Either party has not answered after 30s
b) Either party has gone ON-HOOK
c) Either party has operated the telephone cradle 3 times to call the operator on an established call.

d) Time Messages 3 Types are given if there are no LED type messages.

1) e.g. "TIME IS TWELVE FIFTY TWO" for an ordinary time announcement when the Footswitch is pressed.

2) "SET HOURS" followed by an incrementing hour number if the "Set Hours" switch is operated.

2) "SET MINUTES" followed by an incrementing minute number if the "Set Minutes" switch is operated.

e) Internal Messages 2 Types are given after a CPU restart.

1) A high tone followed by a low tone if the software successfully completes the hardware test after the Blind Operator Aid is switched on or reset.

2) "DANGER" if the hardware is found to be faulty after the Blind Operator Aid is switched on or reset.

The priorities of the LED messages were selected as:-

1) Cord Circuit Messages: These have the highest priority since there are only 17 Cord Circuits, it being desirable to free these for use as soon as possible to expedite new calls.

2) Exchange Messages: These have the middle Call Message priority, since it is usually desirable to answer the outside caller before a staff- member (unless he happens to be The Chairman of the Board !!).

3) Extension Messages: The internal Extension calls have the lowest priority of the Call Messages.

4) Time Messages: These have lowest priority of any Message type, being used to signal the operator that there are no other Messages relating to calls to be processed.

c) Procedure STATES This Procedure performs actions dependant upon the sequential states of pairs of Sup'y LEDs. For example, if both Sup'y LEDs were ON and then one went OFF, it would be necessary to start the process of timing the Loop Waiting State. There are 16 possible sets of actions corresponding to the 16 combinations of the 4 Sup'y LED states e.g.:-

OFF-OFF to OFF-OFF, OFF-OFF to OFF-ON and so on all the way to the final combination ON-ON to ON-ON.

It is possible, however, to rationalise the consequences of the combinations to find common sets of actions and thus reduce the complexity of Procedure STATES as shown overleaf.

3) State 3 This State is similar to State 2 except that it does not initialise the variable "Wait Limit".

4) State 4 This State is entered for the combinations OFF-ON-OFF-ON and ON-OFF-ON-OFF and is the timing process for checking whether the Cord Circuit should be noted as Waiting.

5) State 5 This State is entered for the combination ON-ON-ON-ON and is the timing process for checking whether the Cord Circuit should be noted as having a call in progress i.e. both Sup'y Indicators ON. If so, then all messages for that Cord Circuit are removed from the Message Stack until further notice!.

Up to the time of writing, the 5 States above have been found to be adequate on the field trial units.

Conclusion

The main purpose of the investigation was to produce a unit that would enable Blind Switchboard Operators to use the standard SAPO Floor Pattern Switchboard.

The author considers that along the way he achieved the following:-

a) A flexible microprocessor-controlled hardware design that allows the ergonomics and subjective relationship between the operator and the unit to be explored by software changes only. For example:-

1) The provision of two "Call Unanswered" timeouts:-

a) One of 25s after the first time a call is accepted on the Switchboard, allowing the operator approximately 10s to set up the call, leaving 15s for the called party to answer.
b) 15s timeouts each time the operator is subsequently alerted that the called party has not answered.

The subjective effect of the above is that the caller is serviced at approximately equal intervals of 15s. The 15s timeout was chosen by the author from experience gained with a similar facility on an electronic digital PEBX.

2) By providing an easily changed message format, the words spoken by DIGITALKER can be repeated or given once only and can even be altered to suit various operators.

b) Interfacing between the Floor Pattern Switchboard and the Blind Operator Aid that will work with all Floor Pattern Switchboards, both those in production and those installed in the field. This is essential to cope with component variations that occur in mass production.

c) An easily changed programme structure that relates stimulus and response by means of Table Entries and sub-Procedures performing simple tasks.

d) Four field-trial units to help the author evaluate the programme and hardware design (hopefully with four willing operators).

e) A better understanding of Speech Synthesis in its various forms such as PCM, Phoneme Addition and Data Compression.

f) A "Wish List" for the future.

Some items on the "Wish List" which point to future work to be done are:-

a) More investigation into the OKI ADPCM module to produce Afrikaans and application oriented vocabularies.

b) More work on the software to produce dynamically assigned Message Priorities such as:-

1) VIPs, e.g. The Chairman of The Board, would be recognised by Extension Number and given top priority.
2) The number of messages on any stack (such as the Extension, Exchange or Call Circuit stacks) will be taken into consideration rather than merely the priority fixed in ROM.

c) The construction of a 24V PSU to allow connection of the Blind Operator Aid to the Floor Pattern Switchboard battery supply, if fitted (not all installations have battery supplies).

d) The ability to select a 12 or 24 hour Real Time Clock (using the Set Time switches operated simultaneously).

b) The Message Stacks were re-dimensioned in the software to accommodate messages for every Extension, Exchange and Cord Circuit on a fully equipped (20 + 100) Switchboard, allowing strict chronological storage of every activity. Since the software allows only one Message per Indicator, no Messages are lost i.e. there can be no Message overflow.

A final consideration is operator overload due to a high call arrival rate. Since there are only 17 Cord Circuits on the Switchboard, the operator can handle only 17 simultaneous calls (whether he is Blind or Sighted). Thus any further incoming calls will have to wait until a Cord Circuit becomes free. One major advantage of the Blind Operator Aid is that, since the Messages are stacked in chronological order, the operator provides a true "First come, first served" answering service.

The author thanks the reader for staying with him 'til the end of this chapter.

FIN

Appendix

A.1 Diag. Notes Relating to LED Electronic Extension Indicator

Circuit Diagram A.1.1 Overleaf Refers

Facilities

1) Provides a battery feed to the associated extension while signalling the operator.

2) When the extension lifts the telephone, the common switchboard pilot circuit is energised (to sound the audible warning device if required) and the LED on the extension indicator lights.

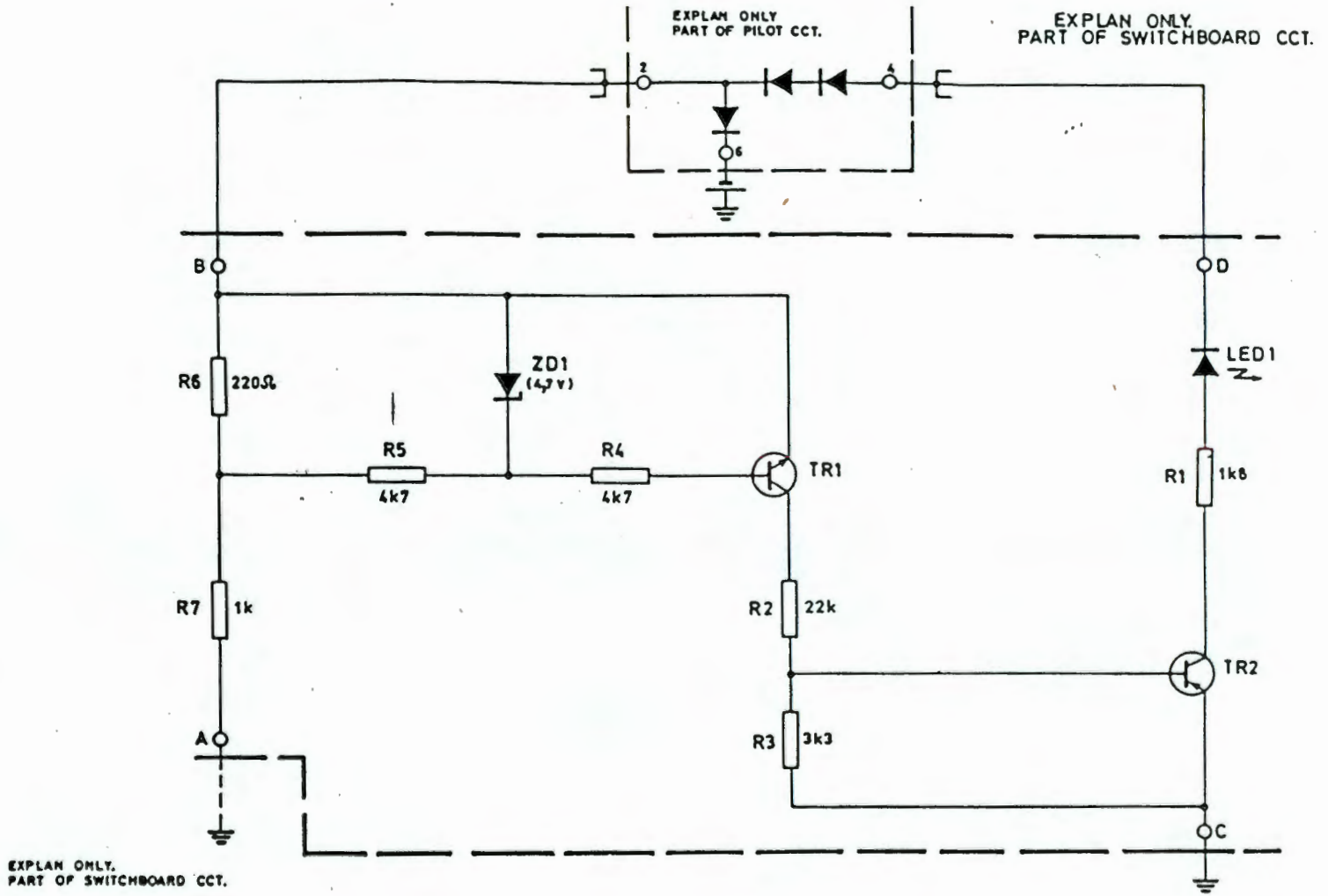
Circuit Description

The -24V battery is connected via the pilot circuit to terminal B. Terminal A is connected to one leg of the extension telephone, the other leg of the telephone being earthed.

When the line is looped, by lifting the telephone handset, current flows through R6 and R7. The voltage drop across R6 is sufficient to turn on TR1 via R5 and R4. TR1, via R2 and R3, turns on TR2 which lights LED1 via R1 and the pilot circuit. The pilot circuit responds to this current drain, operating the audible warning device/buzzer/extension bell as required.

ZD1 limits TR1 base current via R4.

END



COMPONENTS	REMARKS	EQUIV
RESISTORS GENERALLY R1 R7	¼ W ⅓ W 2½ W	
DIODES ZENER ZD1 LIGHT EMITTING LED1	1N4732 GPL 173	 5082 - 4658
TRANSISTORS TR1 TR2	 BC 237A BC 307A	 BC 547A BC 557A

A.1.2 Indicator Output Analysis

The Switchboard has an 18V standby supply (2 small series- connected PM9 batteries) fed via D1 on the Pilot Circuit to the Extension Indicators to power the telephone, calling state only. This supply is only rarely used, being adequate to power the Indicators, Pilot Circuit and AWD during a 24 hour mains failure, working down to 14V.

Examining A.1.1, A.2.1 and A.4.1 output circuits, the following Indicator output currents are calculated using the parameters obtained from data sheets:-

$V_{bat\ min} = 14V$ $\qquad$ $V_{LED\ max} = 2,2V$

$V_{sat} = 60mV$ at $I_c = 10mA$ $\qquad$ $V_{ak\ diode} = 0,68V = V_{be}$ to a first approximation

Resistor tolerances = 2%

Extension Indicator (A.1.1)

$$I_{LED\ min} = (V_{bat\ min} - V_{sat} - 2V_{be} - V_{LED\ max})/R1$$

$$= 5,7mA \quad \text{.....(1)}$$

(1) Assumes that TR1 in the Pilot Circuit just turns on.

Exchange Indicator (A.2.1

The total current includes that through the opto-coupler etc. and is thus greater than that in (1) above.

Sup'y Indicator (R4, A.4.1)

Does not include V_{sat} or V_{LED}, thus the current is greater than in (1) above.

Thus, the minimum current into R1 of the Pilot Circuit from any single Indicator type is 5.7mA. This value is used in the analysis of the Scanner Interface (Chapter 4.2.1).

A.2 Diagram Notes Relating to LED Electronic Exchange Indicator

Circuit Diagram A.2.1 Overleaf Refers

Facilities

1) Provides a floating, high resistance input circuit so as not to load the exchange line during incoming ringing.

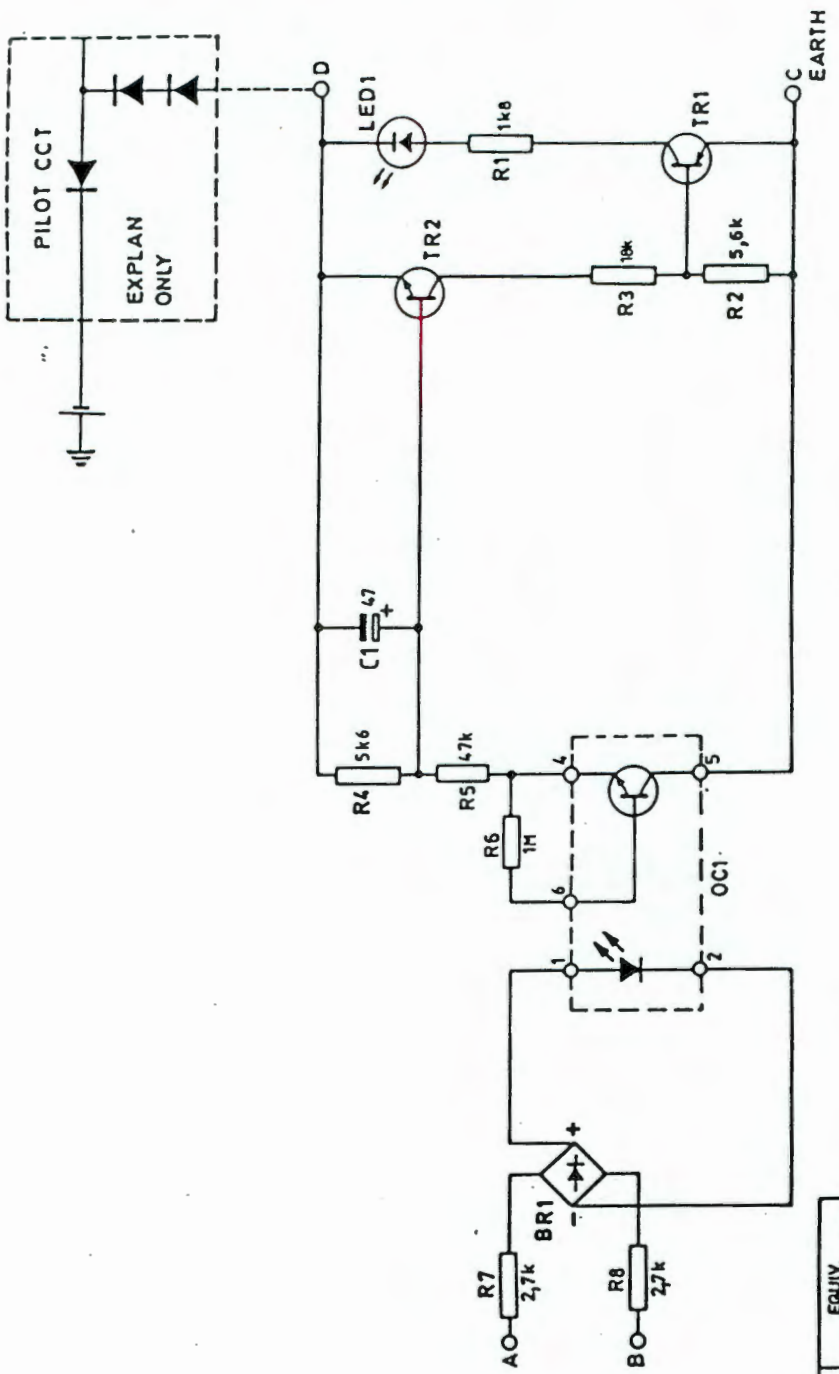
2) When exchange "ring-current" is applied to the input, via a DC isolating 1uF capacitor in the switchboard circuitry, the common pilot circuit is energised (to sound the audible warning device if required) and the lamp on the exchange indicator lights.

CIRCUIT DESCRIPTION

The incoming "ring-current", applied to BR1 via R7 and R8, is rectified and applied to the input of the opto-coupler, OC1. The peaks of this input signal cause the output transistor of OC1 to conduct, which via R5 and R4 turn on TR2. C1 discharges slowly via R4 and TR2 base, thereby smoothing the signal. TR1 thus turns on continuously, via R3 and R2, during each period of ring-current, lighting LED1 via R1 and the pilot circuit. The pilot circuit responds to this current drain, operating the audible warning device/buzzer/extension bell as required.

END

A.2.1 Circuit Diag. for LED Electronic Exchange Indicator



COMPONENTS	REMARKS	EQUIV.
RESISTORS GENERALLY R1 R7, R8	¼ W ½ W 2½ W	
CAPACITOR C1	3V	
DIODES LIGHT EMITTING LED1	GPL 173 7-440-00005-012	5082-4658
RECTIFIER BRIDGE BR1	W 04	
TRANSISTORS TR1 TR2	BC 307 A BC 237 A	BC 557 A BC 547 A
OPTO-COUPLER OC1	MCT 2	

A.3 Diagram Notes Relating to LED Electronic Sup'y Indicator

Circuit Diagram A.3.1 Overleaf Refers

Facilities

1) Provides a low-resistance, electrically symmetrical input circuit with a high resistance to earth. Since the indicator will be in series with the exchange or extension circuit and the telephone, a low voltage drop and independence of line polarity is required.

2) Incorporates an LED which lights when a current greater than 10 mA flows through the input.

3) Provides an output circuit that can be connected in series with that of a second Supervisory Indicator, thereby driving an external relay, (called relay D in the switchboard), only when both Supervisory Indicators are ON.

CIRCUIT DESCRIPTION

In use, the input is connected in series with one leg of the telephone line. Line current flows through R2, with the maximum voltage drop limited by ZD1 and ZD2. When sufficient line current flows, TR1 (or TR2, depending on line current polarity) turns on via R1, applying the line-to-earth voltage to R3, R4 and R8. The voltage across R8, via R11, turns on TR3, which via R5 and R7 turns on TR4 and TR5. This lights LED1 and via R10 turns on the output transistor TR6 (if the emitter has a path to earth).

Each pair of Sup'y Indicator output terminals are connected in series as follows: Sup'y 1C to earth, Sup'y 1D to Sup'y 2C, Sup'y 2D via the D relay to -24V. Thus both indicators must be driven for the D relay, associated with the switchboard circuitry, to operate.

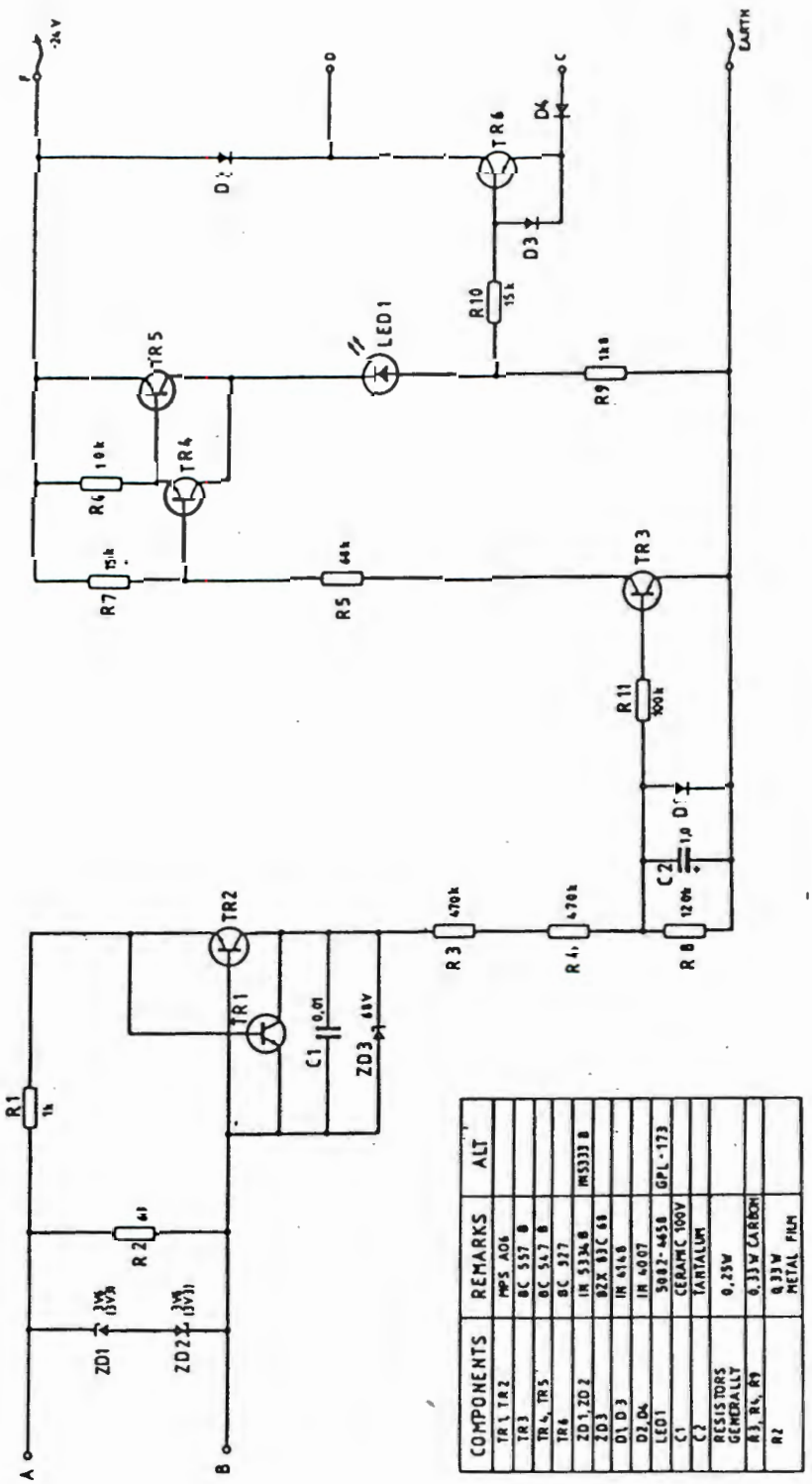
The D relay in turn operates another relay, called the SA relay, which has a self-holding contact. This SA relay remains held, irrespective of the D relay, until both the Cords are removed from the sockets (the earth connection to the SA relay is completed by an isulated mechanical switch on the Cord sockets, these contacts operating in parallel).

Thus, either Sup'y LED going out allows the D relay contact to sound the AWD, via a second SA relay contact, until both Cords are removed.

ZD3 protects TR1 and TR2 against excess line voltages. C1, C2 and D1 prevent 50 Hz signals on the line from affecting operation, D2 clamps the back-e.m.f. of the relay, D3 and D4 prevent damage or false operation from any spurious negative voltages applied to the C terminal.

END

A.3.1 Circuit Diagram for LED Electronic Supply Indicator



COMPONENTS	REMARKS	ALT
TR1, TR2	MPS A06	
TR3	BC 557 B	
TR4, TR5	BC 547 B	
TR6	BC 327	
ZD1, ZD2	IN 5334B	MS333B
ZD3	BZX 83C 6V8	
D1 D3	IN 4148	
D2 D4	IN 4007	
LED1	5082-4458	GPL-173
C1	CERAMIC 100V	
C2	TANTALUM	
RESISTORS GENERALLY	0.25W	
R3, R4, R9	0.33W CARBON	
R2	0.33W METAL FILM	

A.4 Diagram Notes Relating to Pilot Circuit

Circuit Diagram A.4.1 Overleaf Refers

Facilities

1) Draws power from the normal -24V supply when available, otherwise from the -18V stand-by battery

2) Feeds this power to the indicators

3) Responds to current being drawn by one or more indicator LEDs, applying power to the audible warning device/buzzer/ external bell.

CIRCUIT DESCRIPTION

Under normal conditions, power from the switchboard nominal -24V supply, fed via D2, is applied directly to the extension indicators and, via R1, to the output terminals of the exchange and extension indicators. When any line is calling, the associated indicator LED lights via R1, producing sufficient voltage drop to turn on TR1, and hence TR2, applying power to the warning device.

Diodes D3 and D5 limit the voltage drop across R1 so that when several indicators are energised, the LEDs do not vary in brightness.

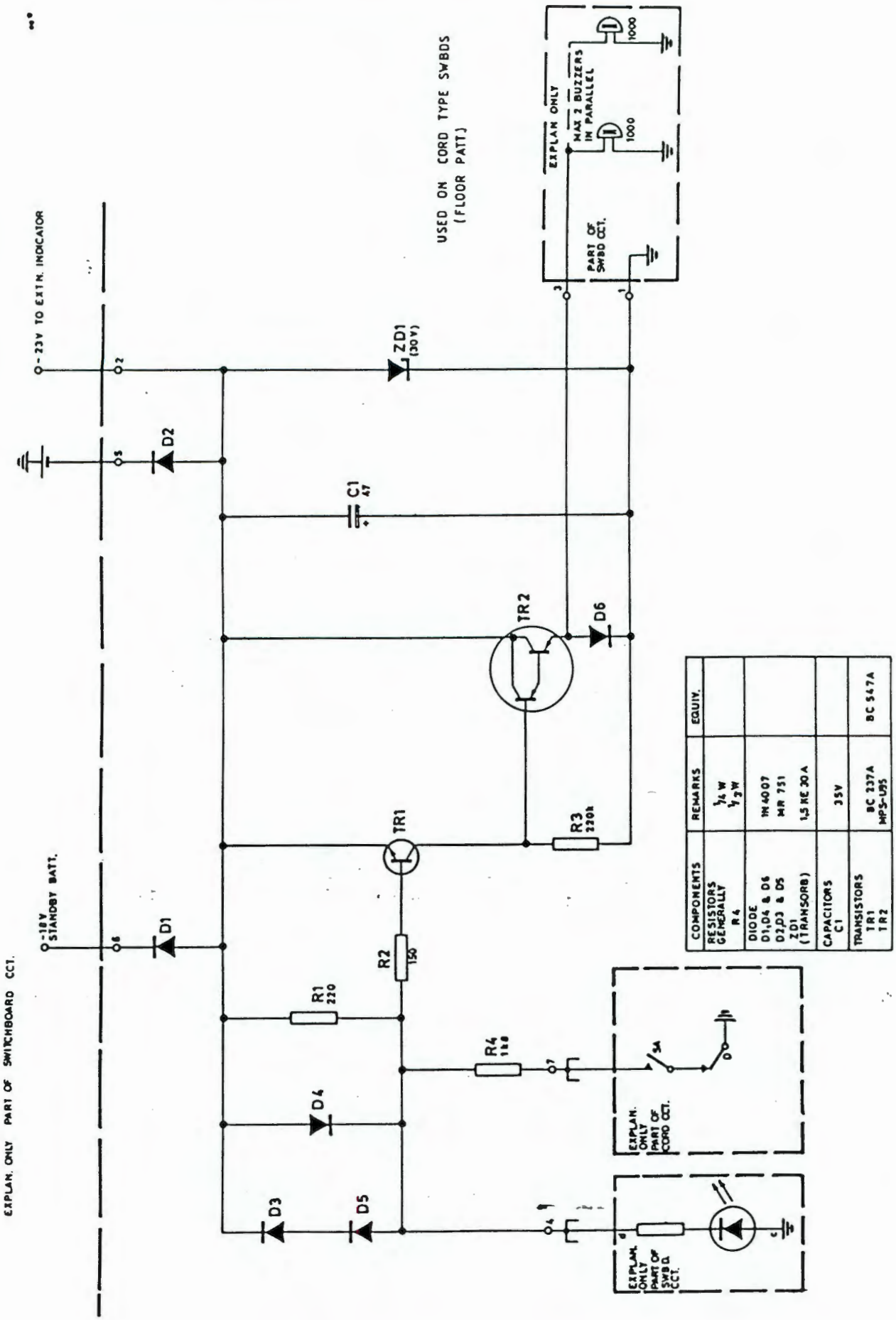
ZD1 is a power zener-diode which limits the maximum voltage that can be induced into the power feeds. D4 and D6 protect against reverse-voltage transients.

In the event of a mains-failure, the -24V mains supply will collapse. The power for the indicators is then supplied by the stand-by battery via D1. D2 prevents the rest of the switchboard from being powered by this battery. C1 provides decoupling at speech frequencies, preventing cross-talk between the various conversations. Cross-talk may occur via the supply rail as the standby battery discharges, becoming relatively high impedance.

R4 provides an input that will accept a full earth signal.

NOTE The AWD sounds only after BOTH Sup'y LEDs have been on and THEN either one goes out (see A.3, Sup'y Indicator).

A.4.1 Circuit Diagram for Pilot Circuit



A.4.2 Pilot Circuit Analysis

Refer to Circuit Diagram A.4.1

The switchboard DC supply, fed to the Pilot Circuit via D2, can be 28V with the batteries fully charged or with the mains-supply set at maximum specified voltage. The AWD, plus two buzzers or bells, is driven by Darlington transistor TR2, the maximum specified load being 60mA.

Using the following parameters, taken from device data sheets, the Pilot Circuit input can be analysed over the ambient temperature range of the Floor Pattern Switchboard (0°C to 50°C):-

h_{fe2} = 25 000 min.(Darlington)
V_{be1} max = 0,62V } at I_c=100µA
V_{be1} min = 0,53V } and 25°C
$V_{be\ D2} = V_{ak\ D2}$ = 0,68V
Resistors are 2% tolerance.

h_{fe1} = 62 min. } at I_c=100µA and 0°C
V_{sat1} = 0,05V at I_c=100µA
$|V_{bat\ max}|$ = 28V

I_{b1} needed to saturate TR1 and hence drive TR2 fully is calculated below. The term V_{be} takes into account D2 in series with V_{bat}:-

$$I_{b1} = 10((V_{bat}-V_{be}-V_{sat})/R3 + I_{c2}/h_{fe2})/h_{fe1}$$
$$= 20\mu A, \text{ allowing a safety factor of 10 for good saturation of TR1} \quad .....(1)$$

To turn TR1 on, the voltage across R1 must be:-

$$V_{R1} = I_{b1}*R2 + V_{be\ 25^o} + dV_{be}/d_T*(T_a-25) \quad .....(2)$$

Using dV_{be}/d_T of -2,2mV/°C and V_{be} limits at 25°C as above at the temperature extremes:-

$V_{R1\ max}$ at 0°C = 660mV .....(3)
$V_{R1\ min}$ at 50°C = 480mV .....(4)

A.5 CMOS Input Analysis

The following analysis calculates the maximum values of the resistors that can be used for the potential divider on the input of a digital CMOS device to ensure that true logic levels are detected as with the following inputs conditions:-

a) With V_{in} = open-circuit, a logic "0" is sensed.
b) With V_{in} = (V_{cc} - 0,2V), a logic "1" is sensed.

The following parameters are from CMOS device data sheets:-

$I_{in\ max}$ = 0,3µA, source and sink, with V_{cc} = +5V
$V_{in\ low}$ = 1,5V max.
$V_{in\ high}$ = 3,5v min.

The values of R_X and R_Y, Fig.4.1 in Chapter 4, are calculated below:-

For the case of an open-circuit input:-

$$R_Y = V_{in\ low} / I_{in\ max}$$

$$= 5M\Omega$$

Take R_Y = 4,7MΩ, ±2% .....(1)
as the nearest lower standard value

For the case of Fig.4.1, where R_X is connected to a saturated Indicator transistor i.e. (V_{cc} - V_{sat}):-

$$R_X = (V_{cc} - V_{sat} - V_{in\ high}) / (V_{in\ high} / R_{Y\ min} + I_{in\ max})$$

$$= 1,32M\Omega$$

Take R_X = 1,2MΩ, ±2% .....(2)
as the nearest lower standard value

The values of R_X and R_Y calculated are above used in the further analysis of the Scanner Interface (Chapter 4.2.1).

A.6 Notes on Plunger, Drop-flap and Eyeball Indicators

a) Plunger Tactile Indicator (A.5.1.a)

These Indicators are mounted vertically.

Current through the coil draws the soft-steel Armature up towards the mounting plate, causing the button (plunger) to protrude fully. When current stops flowing, the button and armature fall to the rest position due to gravity.

When driven by electronic circuitry, the coil current is usually chopped at approximately 30Hz. The resultant vibration in the button aids tactile sensing of active Indicators.

b) Drop-flap Indicator (A.5.1.b)

These Indicators are mounted horizontally.

Current through the coil attracts the Armature towards the Pole-piece, lifting the Armature-rod and releasing the Shutter (Drop-flap), which then rotates to a horizontal position due to gravity.

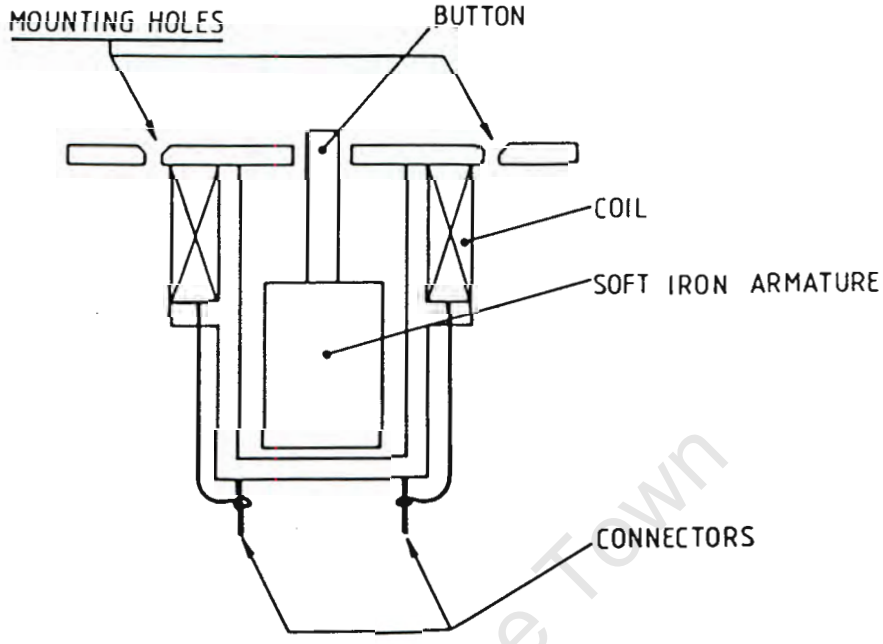
The operator restores the Shutter by hand.

c) Eyeball Indicator (A.5.1.c)

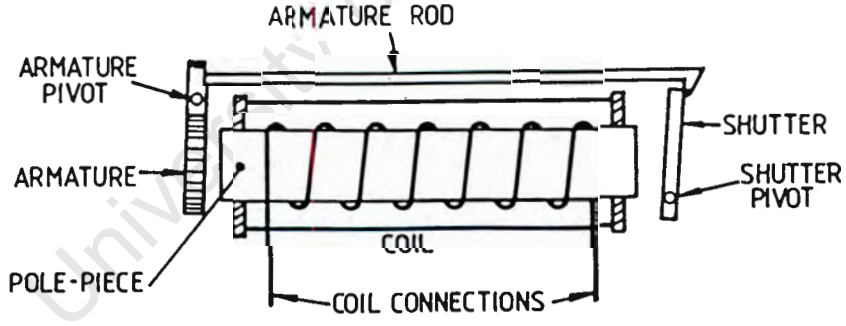
These Indicators are mounted horizontally.

Current through the coil draws the Armature, hanging vertically down at rest due to gravity, towards the Pole-piece, thereby rotating the Eyeball disk to appear in the Aperture (shown dotted). The Eyeball is usually painted white or yellow.

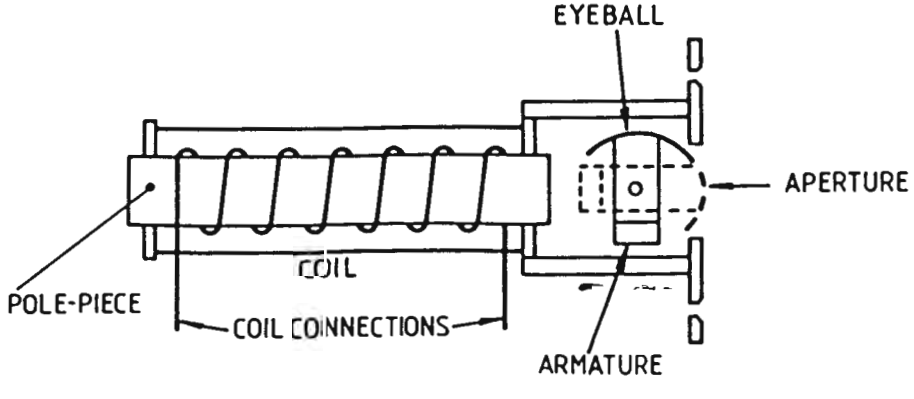
A.6.1 Drawings of Plunger, Drop-flap and Eyeball Indicators



a)

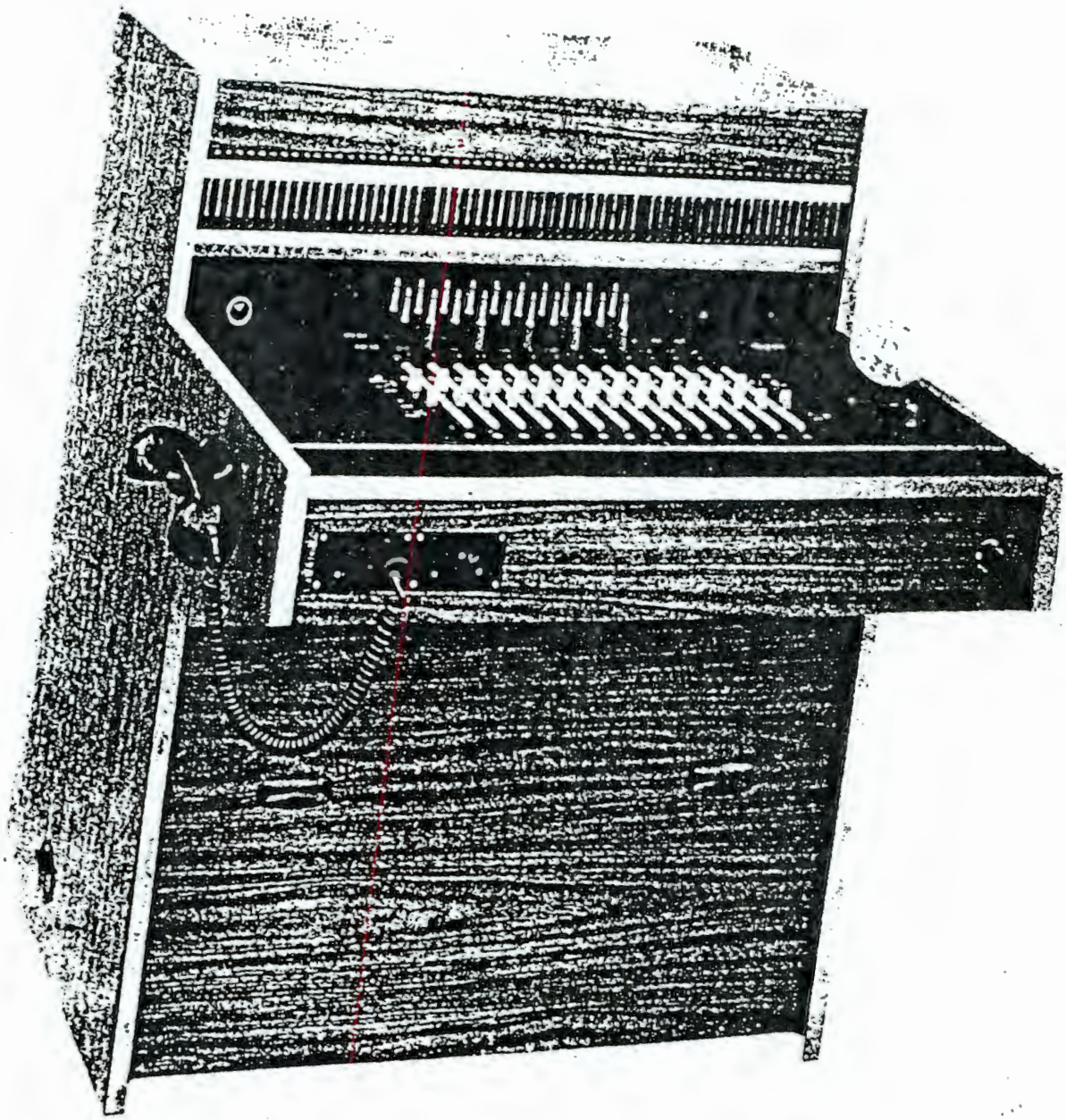


b)

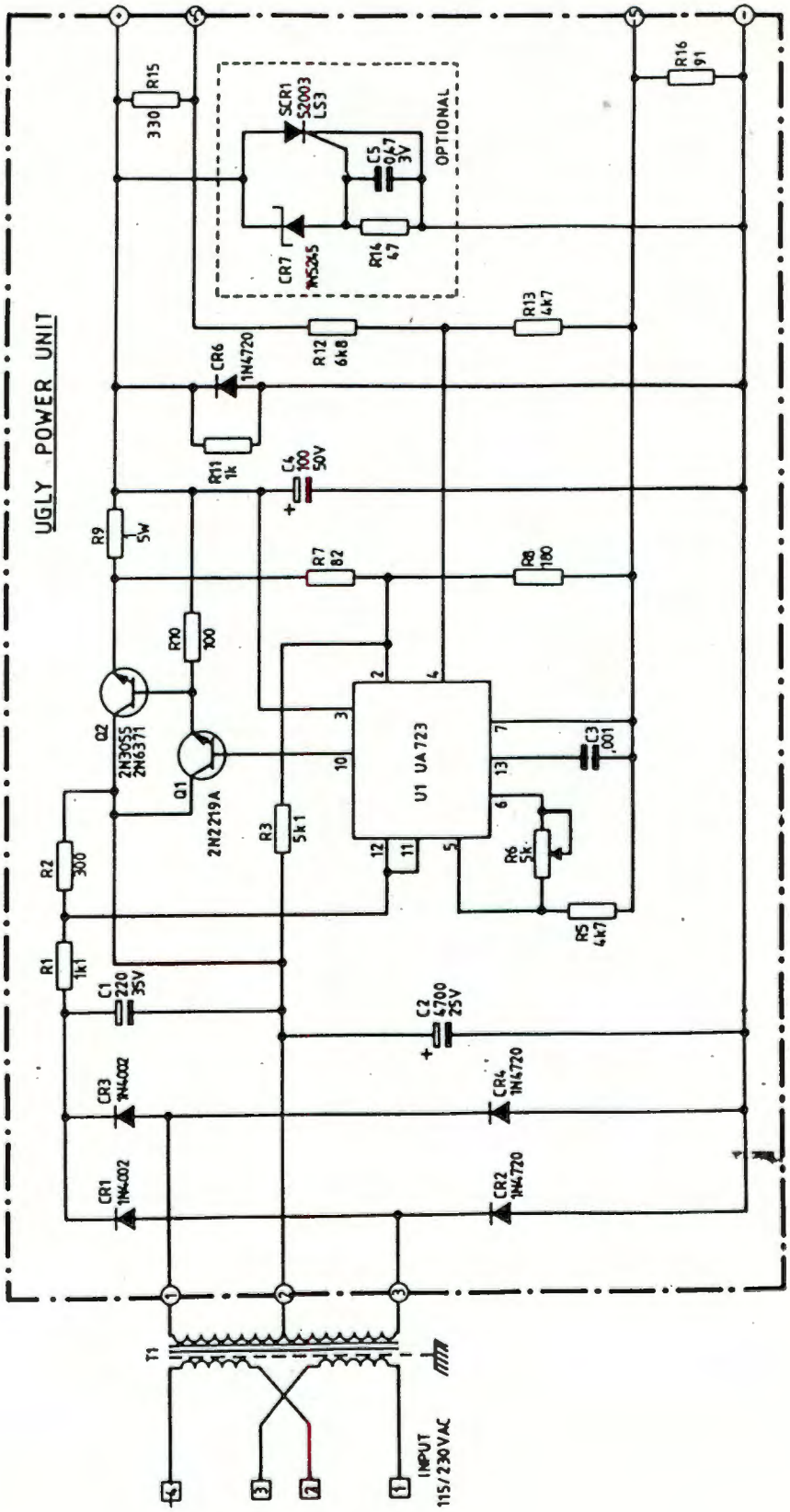


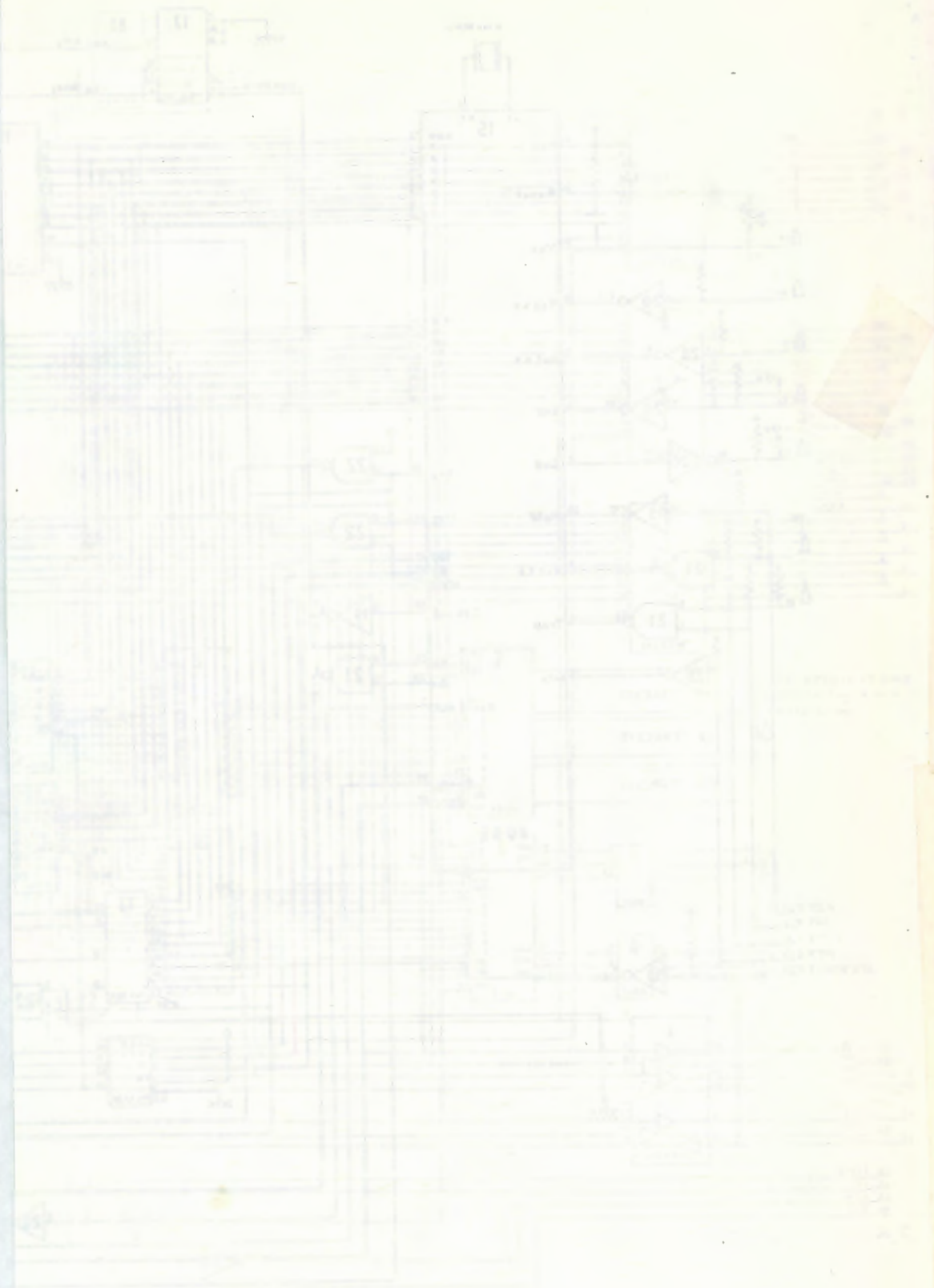
c)

A.7 Picture of Tadiran Electronic Industries Audible FPS



A.8 Circuit Diagram of Mains-to-12V PSU





A.13 Bibliography

Software:-

1) Kernighan and Plauger
"The Elements of Programming Style"

2) De Pledge, Dr
"Course Notes for Microprocessor Software"
U.C.T. Post-Graduate Program

3) Intel Corporation
"8085 Assembly Language Programming Manual"

Hardware:-

4) National Semiconductor Corporation
"CMOS Data Book"

5) National Semiconductor Corporation
"Audio Data Book"

6) National Semiconductor Corporation
"TTL Data Book"

7) Intel Corporation
"8085 System User's Manual"

8) Atkinson J
"Telephony, Vol 1"

9) OKI Electric Industry Co Ltd.
"OKI RealVoice Series" Data Book

10) Plessey South Africa Limited
"Various Equipment Manuals"

Bibliography Continued

11) CCITT
"Recommendations G732 & G711 for PCM"

12) Ciarcia S
"Use ADPCM for Highly Intelligible Speech"
Byte Magazine.

13) Bennet G H
"PCM and Digital Transmission"
Marconi Press.

A.14 Software Listings

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                        1 "8085"
                        3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                        4  EXPAND
                        5 ************************************************************
                        6 *                                                          *
                        7 * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.    *
                        8 *                                                          *
                        9 * PRODUCT CODE:   9-XXX-XXXXX-XXX                         *
                       10 *                                                          *
                       11 * MODULE:         MAIN                                   *
                       12 *                                                          *
                       13 * FUNCTION:       CALLS SCANNING , VOICE OUTPUT , BACKGROUND *
                       14 *                 ROUTINES                                 *
                       15 *                                                          *
                       16 * CALLS:          LMPSCN,LMPSTT,SRTEXT,SRTEXC,SRTLOP,OUTVCE *
                       17 *                 GENMSG,TIMER,TIMESET                     *
                       18 *                                                          *
                       19 * ON TAPE:        NOT YET                                  *
                       20 *                                                          *
                       21 * LINKING:        LINK BLIND                               *
                       22 * INSTRUCTIONS                                            *
                       23 *                                                          *
                       24 * FILE HISTORY:   STARTED 8/7/82                          *
                       25 *                                                          *
                       26 * 8085 CONVERSION STARTED 16/09/83                        *
                       27 *                                                          *
                       28 * 0.0 08/07/82    INITIAL CONCEPT                         *
                       29 *                                                          *
                       30 *                                                          *
                       31 ************************************************************
                       32
                       33          GLOBAL              VOICE,PORTA,PORTB,PORTC
                       34
                       35          EXTERNAL            LMPSCN,LMPSTT,LOPSTT,SRTEXT,SRTEXC,SRTLOP
                       36          EXTERNAL            OUTVCE,GENMSG,TIMER,TIMSET
                       37          EXTERNAL            EXTMSTK,EXCMSTK,LOPMSTK,STTSTK
                       38          EXTERNAL            HITONE,LOWTONE,BUFFER,ENDMSG,OFFLG
                       39
                       40                              ;VAR:
                       41                              ;       WTCHDG  : BYTE;
                       42                              ;       VOICE   : BYTE;
                       43                              ;       DIGEN   : BYTE;
                       44                              ;       DISABL  : BYTE;
                       45                              ;
                       46                              ;CONST:
<00EF>                 47 WTCHDG  EQU   11101111B      ;       @WTCHDG = 11101111B;
<00CF>                 48 VOICE   EQU   11001111B      ;       @VOICE  = 11001111B;
<00DF>                 49 DIGEN   EQU   11011111B      ;       @DIGEN  = 11011111B;
<00FF>                 50 DISABL  EQU   11111111B      ;       @DISABL = 11111111B;
<00A0>                 51 PORTA   EQU   0A0H           ;       @PIA PORT A = 0A0H ;
<00A1>                 52 PORTB   EQU   0A1H           ;       @PIA PORT B = 0A1H ;
<00A2>                 53 PORTC   EQU   0A2H           ;       @PIA PORT C = 0A2H ;
<00A3>                 54 CONTRL  EQU   0A3H           ;       @PIA CONTROL ;
<1000>                 55 RAMBOT  EQU   01000H         ;       @RAMBOT = 01000H;
<13FF>                 56 RAMTOP  EQU   013FFH         ;       @RAMTOP = 013FFH;
<0000>                 57 ROMBOT  EQU   00000H         ;       @ROMBOT = 00000H;
<0FFF>                 58 ROMTOP  EQU   ROMBOT+0FFFH   ;       @ROMTOP = ROMBOT+0FFFH;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

          <0000>     59 ZERO     EQU     000H        ;       ZERO    = 000H;
          <0000>     60 CLEAR    EQU     000H        ;       CLEAR   = 000H;
          <00FF>     61 SET      EQU     0FFH        ;       SET     = 0FFH;
          <0001>     62 TEST1    EQU     01H         ;       TEST1   = 01H;
          <0002>     63 TEST2    EQU     02H         ;       TEST2   = 02H;
          <004C>     64 DANGER   EQU     4CH         ;       DANGER  = 4CH;
                     65
                     66          DATA                ;VAR:
0000                 67 INTFLG   DS      1           ;       INTFLG  : BYTE;
0001                 68 LOPNUM   DS      1           ;       LOPNUM  : BYTE;
                     69                              ;
                     70                              ;*************************************
                     71                              ;
                     72          PROG                ;PROCEDURE "MAIN";
                     73                              ;   BEGIN
0000 F3              74 MAIN     DI                  ;       DISABLE INTERRUPTS;
                     75                              ;
0001 0664            76          MVI     B,100       ;       LINK TIMLOOP(100 MS.)
0003 05              77 BLOOP    DCR     B           ;           BEGIN
0004 CA0010          78          JZ      ENDTIM      ;
0007 3E8D            79          MVI     A,141       ;
0009 3D              80 ALOOP    DCR     A           ;
000A C20009          81          JNZ     ALOOP       ;
000D C30003          82          JMP     BLOOP       ;
0010 00              83 ENDTIM   NOP                 ;           ENDLINK;
                     84                              ;
0011 3113FF          85          LXI     SP,RAMTOP   ;       STKPTR = RAMTOP;
0014 3E83            86          MVI     A,83H       ;
0016 D3A3            87          OUT     CONTRL      ;       CONFIGURE PIA;
0018 3EDF            88          MVI     A,DIGEN     ;
001A D3A2            89          OUT     PORTC       ;
001C 3E00            90          MVI     A,HITONE    ;
001E D3CF            91          OUT     VOICE       ;               VOICE = HITONE;
0020 3EFF            92          MVI     A,DISABL    ;
0022 D3A2            93          OUT     PORTC       ;
                     94                              ;
0024 0601            95          MVI     B,01H       ;
0026 211400          96          LXI     H,RAMTOP+1  ;       FOR PTR = RAMTOP TO RAMBOT DO
0029 2B              97 LOOP2    DCX     H           ;           BEGIN
002A 70              98 LOOPL2   MOV     M,B         ;
002B 2B              99          DCX     H           ;
002C 111000         100          LXI     D,RAMBOT    ;
002F 7C             101          MOV     A,H         ;
0030 BA             102          CMP     D           ;
0031 C2002A         103          JNZ     LOOPL2      ;
0034 7D             104          MOV     A,L         ;
0035 BB             105          CMP     E           ;
0036 C2002A         106          JNZ     LOOPL2      ;
0039 70             107          MOV     M,B         ;
003A 78             108 LOOPL3   MOV     A,B         ;
003B BE             109          CMP     M           ;
003C C20055         110          JNZ     ENDFR2      ;               LINK RAMTST
003F 36FF           111          MVI     M,0FFH      ;
0041 23             112          INX     H           ;
0042 111400         113          LXI     D,RAMTOP+1  ;
0045 7C             114          MOV     A,H         ;
0046 BA             115          CMP     D           ;
```

(123)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0047 C2003A       116         JNZ   LOOPL3      ;
    004A 7D           117         MOV   A,L         ;
    004B BB           118         CMP   E           ;
    004C C2003A       119         JNZ   LOOPL3      ;
    004F 78           120         MOV   A,B         ;
    0050 17           121         RAL               ;
    0051 47           122         MOV   B,A         ;
    0052 D20029       123         JNC   LOOP2       ;
    0055 00           124 ENDFR2  NOP               ;         ENDFOR;
                      125                           ;
    0056 111400       126         LXI   D,RAMTOP+1  ;     IF PTR <> (RAMTOP + 1) OR TESTPAT <> ZERO THEN
    0059 7C           127         MOV   A,H         ;
    005A BA           128         CMP   D           ;
    005B C20069       129         JNZ   NOTEQU      ;
    005E 7D           130         MOV   A,L         ;
    005F BB           131         CMP   E           ;
    0060 C20069       132         JNZ   NOTEQU      ;
                      133                           ;
    0063 78           134         MOV   A,B         ;
    0064 FE00         135         CPI   ZERO        ;
    0066 CA0078       136         JZ    ELSE4       ;
    0069 3EDF         137 NOTEQU  MVI   A,DIGEN     ;         BEGIN
    006B D3A2         138         OUT   PORTC       ;
    006D 3E4C         139         MVI   A,DANGER    ;
    006F D3CF         140         OUT   VOICE       ;             VOICE = DANGER;
    0071 3EFF         141         MVI   A,DISABL    ;
    0073 D3A2         142         OUT   PORTC       ;
                      143                           ;
    0075 C30085       144 ENDIF4  JMP   ENDLS4      ;         ENDIF;
                      145                           ;
    0078 00           146 ELSE4   NOP               ;     ELSE
    0079 3EDF         147         MVI   A,DIGEN     ;         BEGIN
    007B D3A2         148         OUT   PORTC       ;
    007D 3E00         149         MVI   A,LOWTONE   ;
    007F D3CF         150         OUT   VOICE       ;             VOICE = LOWTONE;
    0081 3EFF         151         MVI   A,DISABL    ;
    0083 D3A2         152         OUT   PORTC       ;
    0085 00           153 ENDLS4  NOP               ;         ENDELSE;
                      154                           ;
    0086 3EEF         155         MVI   A,WTCHDG    ;
    0088 D3A2         156         OUT   PORTC       ;     WATCHDOG = FED;
    008A 3EFF         157         MVI   A,11111111B ;
    008C D3A2         158         OUT   PORTC       ;
                      159                           ;
    008E 0664         160         MVI   B,100       ;     LINK TIMLOOP(100 MS.)
    0090 05           161 BLOOP2  DCR   B           ;         BEGIN
    0091 CA009D       162         JZ    ENDTIM2     ;
    0094 3E8D         163         MVI   A,141       ;
    0096 3D           164 ALOOP2  DCR   A           ;
    0097 C20096       165         JNZ   ALOOP2      ;
    009A C30090       166         JMP   BLOOP2      ;
    009D 00           167 ENDTIM2 NOP               ;         ENDLINK;
                      168                           ;
    009E 211000       169         LXI   H,RAMBOT    ;
    00A1 111400       170 LOOP3   LXI   D,RAMTOP+1  ;     FOR PTR = RAMBOT TO RAMTOP DO
    00A4 7C           171         MOV   A,H         ;
    00A5 BA           172         CMP   D           ;
```

(124)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    00A6 C200AE       173          JNZ     NOTEQ3      ;
    00A9 7D           174          MOV     A,L         ;
    00AA BB           175          CMP     E           ;
    00AB CA00B4       176          JZ      ENDFR3      ;           BEGIN
    00AE 3600         177 NOTEQ3   MVI     M,ZERO      ;               ->PTR = ZERO
    00B0 23           178 NXTFR3   INX     H           ;
    00B1 C300A1       179          JMP     LOOP3       ;
    00B4 00           180 ENDFR3   NOP                 ;           ENDFOR;
                      181                              ;
    00B5 3E01         182          MVI     A,01H       ;
    00B7 320001       183          STA     LOPNUM      ;
    00BA 3A0001       184 FORLOP4  LDA     LOPNUM      ;
    00BD FE12         185          CPI     18          ;       FOR LOPNUM = 1 TO 17 DO
    00BF D200DB       186          JNC     ENDFOR4     ;           BEGIN
    00C2 3D           187          DCR     A           ;
    00C3 47           188          MOV     B,A         ;
    00C4 87           189          ADD     A           ;
    00C5 87           190          ADD     A           ;
    00C6 80           191          ADD     B           ;
    00C7 4F           192          MOV     C,A         ;
    00C8 0600         193          MVI     B,00H       ;
    00CA 210000       194          LXI     H,STTSTK    ;
    00CD 09           195          DAD     B           ;               STATPTR = STTSTK+(5*(LOPNUM-1));
    00CE 3E00         196          MVI     A,OFFLG     ;
    00D0 77           197          MOV     M,A         ;               ->STATPTR = OFFLG
    00D1 3A0001       198 NXTFOR4  LDA     LOPNUM      ;
    00D4 3C           199          INR     A           ;
    00D5 320001       200          STA     LOPNUM      ;
    00D8 C300BA       201          JMP     FORLOP4     ;
    00DB 00           202 ENDFOR4  NOP                 ;           ENDFOR;
                      203                              ;
    00DC 3E00         204          MVI     A,ZERO      ;
    00DE 210000       205          LXI     H,EXTMSTK   ;
    00E1 22FFFE       206          SHLD    EXTMSTK-2   ;       EXTNXTPTR = EXTMSTK;
    00E4 32FFFD       207          STA     EXTMSTK-3   ;       EXTMESNUM = ZERO;
    00E7 210000       208          LXI     H,EXCMSTK   ;
    00EA 22FFFE       209          SHLD    EXCMSTK-2   ;       EXCNXTPTR = EXCMSTK;
    00ED 32FFFD       210          STA     EXCMSTK-3   ;       EXCMESNUM = ZERO;
    00F0 210000       211          LXI     H,LOPMSTK   ;
    00F3 22FFFE       212          SHLD    LOPMSTK-2   ;       LOPNXTPTR = LOPMSTK;
    00F6 32FFFD       213          STA     LOPMSTK-3   ;       LOPMESNUM = ZERO;
    00F9 210000       214          LXI     H,BUFFER    ;
    00FC 22FFFE       215          SHLD    BUFFER-2    ;       BUFPTR = BUFFER;
    00FF 3E00         216          MVI     A,ENDMSG    ;
    0101 320000       217          STA     BUFFER      ;       BUFFER = ENDMSG;
                      218                              ;
                      219                              ;
    0104 3EEF         220          MVI     A,WTCHDG    ;
    0106 D3A2         221          OUT     PORTC       ;       WATCHDOG = FED;
    0108 3EFF         222          MVI     A,11111111B ;
    010A D3A2         223          OUT     PORTC       ;
                      224                              ;
    010C 3E1B         225          MVI     A,00011011B ;
    010E 30           226          SIM                 ;       INTMSK = RESET;
    010F FB           227          EI                  ;       ENABLE INTERRUPT;
                      228                              ;
    0110 00           229 REPEAT   NOP                 ;       REPEAT
```

(125)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                       1  "8085"
                       3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                       4  EXPAND
                       5  ********************************************************************
                       6  *                                                                  *
                       7  * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.           *
                       8  *                                                                  *
                       9  * PRODUCT CODE:   9-XXX-XXXXX-XXX                                 *
                      10  *                                                                  *
                      11  * MODULE:         TIMER                                           *
                      12  *                                                                  *
                      13  * FUNCTION:       THIS MODULE HANDLES THE TIMER THAT OCCURS        *
                      14  *                 EVERY 100 MILLISECOND. "TENTHS" OF A SECOND      *
                      15  *                 IS UPDATED AND USED TO PROVIDE A REAL TIME       *
                      16  *                 CLOCK. "MAIN" IS ALSO SIGNALLED TO DO A          *
                      17  *                 SCAN OF THE LAMPS, UPDATE STATUS ETC.            *
                      18  *                                                                  *
                      19  * CALLS:          NONE                                            *
                      20  *                                                                  *
                      21  * ON TAPE:        NOT YET                                         *
                      22  *                                                                  *
                      23  * LINKING:        LINK BLIND                                      *
                      24  * INSTRUCTIONS                                                     *
                      25  *                                                                  *
                      26  * FILE HISTORY:   STARTED 8/7/82                                  *
                      27  *                                                                  *
                      28  * 0.0 08/07/82    INITIAL CONCEPT                                 *
                      29  *                                                                  *
                      30  *                                                                  *
                      31  ********************************************************************
                      32
                      33          GLOBAL              TIMER,HOURS,MINUTES,SECONDS,TENTHS
                      34
                      35                              ;
                      36                              ;CONST:
                      37                              ;
                      38                              ;TYPE:
                      39                              ;
                      40                              ;
                      41          DATA                ;VAR:
0000                  42 HOURS    DS      1           ;       HOURS   = BYTE;
0001                  43 MINUTES  DS      1           ;       MINUTES = BYTE;
0002                  44 SECONDS  DS      1           ;       SECONDS = BYTE;
0003                  45 TENTHS   DS      1           ;       TENTHS  = BYTE;
                      46                              ;
                      47                              ;
                      48          PROG                ;PROCEDURE "TIMER";
                      49                              ;
0000 3A0000           50 TIMER    LDA     HOURS       ;   BEGIN
0003 FE00             51          CPI     00          ;
0005 C2000D           52          JNZ     ENDIF5      ;       IF HOURS = 00 THEN
0008 3E0C             53          MVI     A,12        ;           BEGIN
000A 320000           54          STA     HOURS       ;               HOURS = 12
000D 00               55 ENDIF5   NOP                 ;           ENDIF;
000E 3A0003           56          LDA     TENTHS      ;
0011 3C               57          INR     A           ;
0012 320003           58          STA     TENTHS      ;       TENTHS  = TENTHS+1;
```

```
LOCATION OBJECT CODE LINE      SOURCE LINE

    0015 FE0A          59          CPI     10          ;
    0017 DA0055        60          JC      ENDIF1      ;       IF TENTHS >= 10 THEN
    001A 3E00          61          MVI     A,00        ;           BEGIN
    001C 320003        62          STA     TENTHS      ;               TENTHS  = 00;
    001F 3A0002        63          LDA     SECONDS     ;
    0022 3C            64          INR     A           ;
    0023 320002        65          STA     SECONDS     ;               SECONDS = SECONDS+1;
    0026 FE3C          66          CPI     60          ;
    0028 DA0054        67          JC      ENDIF2      ;               IF SECONDS  >= 60 THEN
    002B 3E00          68          MVI     A,00        ;                   BEGIN
    002D 320002        69          STA     SECONDS     ;                       SECONDS = 00;
    0030 3A0001        70          LDA     MINUTES     ;
    0033 3C            71          INR     A           ;
    0034 320001        72          STA     MINUTES     ;                       MINUTES = MINUTES+1;
    0037 FE3C          73          CPI     60          ;
    0039 DA0053        74          JC      ENDIF3      ;                       IF MINUTES  >= 60 THEN
    003C 3E00          75          MVI     A,00        ;                           BEGIN
    003E 320001        76          STA     MINUTES     ;                               MINUTES = 00;
    0041 3A0000        77          LDA     HOURS       ;
    0044 3C            78          INR     A           ;
    0045 320000        79          STA     HOURS       ;                               HOURS   = HOURS+1;
    0048 FE0D          80          CPI     13          ;
    004A DA0052        81          JC      ENDIF4      ;                               IF HOURS > 12 THEN
    004D 3E01          82          MVI     A,01        ;                                   BEGIN
    004F 320000        83          STA     HOURS       ;                                       HOURS = 01
    0052 00            84 ENDIF4   NOP                 ;                                   ENDIF;
    0053 00            85 ENDIF3   NOP                 ;                               ENDIF;
    0054 00            86 ENDIF2   NOP                 ;                           ENDIF;
    0055 00            87 ENDIF1   NOP                 ;               ENDIF;
    0056 00            88 END      NOP                 ;   END;
    0057 C9            89          RET                 ;RETURN;
                       90                              ;

Errors=    0
```

(128)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1  "8085"
                      3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                      4  EXPAND
                      5  ********************************************************************
                      6  *                                                                  *
                      7  * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.            *
                      8  *                                                                  *
                      9  * PRODUCT CODE:   9-XXX-XXXXX-XXX                                  *
                     10  *                                                                  *
                     11  * MODULE:         TIMESET                                          *
                     12  *                                                                  *
                     13  * FUNCTION:       THIS MODULE ALLOWS THE OPERATOR TO SET THE       *
                     14  *                 TIME OF DAY TO THE NEAREST MINUTE. THE HOUR      *
                     15  *                 AND MINUTE SET BUTTONS ARE HELD DOWN UNTIL       *
                     16  *                 THE CORRECT HOUR OR MINUTE IS ANNOUNCED.         *
                     17  *                                                                  *
                     18  * CALLS:          BINTOBCD,TFRMSG                                  *
                     19  *                                                                  *
                     20  * ON TAPE:        NOT YET                                          *
                     21  *                                                                  *
                     22  * LINKING:        LINK BLIND                                       *
                     23  * INSTRUCTIONS                                                     *
                     24  *                                                                  *
                     25  * FILE HISTORY:   STARTED 8/7/82                                   *
                     26  *                                                                  *
                     27  * 8085 CONVERSION STARTED 23/09/83                                 *
                     28  *                                                                  *
                     29  * 0.0 08/07/82    INITIAL CONCEPT                                  *
                     30  *                                                                  *
                     31  *                                                                  *
                     32  ********************************************************************
                     33
                     34          GLOBAL              TIMSET
                     35
                     36          EXTERNAL            SPACE20,ZERO,BASE,BINBCD,TFRMSG
                     37          EXTERNAL            VCERDY,PORTB,PORTC,DATSTK
                     38          EXTERNAL            HRSMSG,MINMSG,BCDMSG,VOICE,NEWMSG
                     39          EXTERNAL            TENTHS,SECONDS,MINUTES,HOURS
                     40
                     41                              ;CONST:
<00FF>               42 SET      EQU     11111111B   ;       SET     = 11111111B;
<0000>               43 CLEAR    EQU     00000000B   ;       CLEAR   = 00000000B;
<0002>               44 HRSMSK   EQU     00000010B   ;       HRSMSK  = 00000010B;
<0004>               45 MINMSK   EQU     00000100B   ;       MINMSK  = 00000100B;
<0000>               46 DATA     EQU     PORTB       ;       @DATA   = PORTB;
<00DF>               47 DIGEN    EQU     11011111B   ;       DIGEN   = 11011111B;
<00FF>               48 DISABL   EQU     11111111B   ;       DISABL  = 11111111B;
                     49                              ;
                     50                              ;
                     51                              ;TYPE:
                     52                              ;       DATA    : BYTE;
                     53                              ;
                     54          DATA                ;VAR:
0000                 55 DATPTR   DS      2           ;       DATPTR  = PTR;
0002                 56 HRSFLG   DS      1           ;       HRSFLAG = BYTE;
0003                 57 MINFLG   DS      1           ;       MINFLAG = BYTE;
                     58                              ;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      59          PROG                ;PROCEDURE "TIMESET";
                      60                              ;
    0000 3A0000       61 TIMSET   LDA     TENTHS      ;   BEGIN
    0003 FE00         62          CPI     00          ;
    0005 CA000D       63          JZ      NEXT1       ;
    0008 FE05         64          CPI     05          ;
    000A C2001F       65          JNZ     ELSE01      ;
    000D DB00         66 NEXT1    IN      DATA        ;
    000F E600         67          ANI     VCERDY      ;
    0011 CA001F       68          JZ      ELSE01      ;
    0014 3A0000       69          LDA     NEWMSG      ;
    0017 FE00         70          CPI     CLEAR       ;
    0019 C2001F       71          JNZ     ELSE01      ;       IF TENTHS = (00 OR 05) AND VOICE = READY AND NEWMSG = CLEAR THEN
    001C C30022       72          JMP     CONT1       ;
    001F C3011F       73 ELSE01   JMP     ELSE1       ;
    0022 DB00         74 CONT1    IN      DATA        ;           BEGIN
    0024 E602         75          ANI     HRSMSK      ;
    0026 C20075       76          JNZ     ELSE2       ;
    0029 DB00         77          IN      DATA        ;
    002B E604         78          ANI     MINMSK      ;
    002D CA0075       79          JZ      ELSE2       ;               IF SETHRS = SET AND SETMIN = CLEAR THEN
    0030 3A0002       80          LDA     HRSFLG      ;
    0033 FE00         81          CPI     CLEAR       ;                   BEGIN
    0035 C20046       82          JNZ     ELSE3       ;                       IF HRSFLG = CLEAR THEN
    0038 210000       83          LXI     H,HRSMSG    ;                           BEGIN
    003B CD0000       84          CALL    TFRMSG      ;                               "GO TRANSFER MESSAGE"(,,,,HRSMSG);
    003E 3EFF         85          MVI     A,SET       ;
    0040 320002       86          STA     HRSFLG      ;                               HRSFLG = SET
    0043 C30071       87 ENDIF3   JMP     ENDLS3      ;                           ENDIF;
    0046 00           88 ELSE3    NOP                 ;                       ELSE
    0047 3E00         89          MVI     A,00        ;                           BEGIN
    0049 320000       90          STA     SECONDS     ;                               SECONDS = 00;
    004C 3A0000       91          LDA     HOURS       ;
    004F C601         92          ADI     01          ;
    0051 320000       93          STA     HOURS       ;                               HOURS = HOURS+1;
    0054 FE0D         94          CPI     13          ;
    0056 DA005E       95          JC      ENDIF4      ;                               IF HOURS > 12 THEN
    0059 3E01         96          MVI     A,01        ;                                   BEGIN
    005B 320000       97          STA     HOURS       ;                                       HOURS = 01
    005E 00           98 ENDIF4   NOP                 ;                                   ENDIF;
    005F 3EDF         99          MVI     A,DIGEN     ;                               ELSE
    0061 D300        100          OUT     PORTC       ;
    0063 3A0000      101          LDA     HOURS       ;
    0066 D300        102          OUT     VOICE       ;                               VOICE = HOURS;
    0068 3EFF        103          MVI     A,DISABL    ;
    006A D300        104          OUT     PORTC       ;
    006C 3EFF        105          MVI     A,SET       ;
    006E 320000      106          STA     NEWMSG      ;                               NEWMSG = SET
    0071 00          107 ENDLS3   NOP                 ;                           ENDELSE;
    0072 C3011B      108 ENDIF2   JMP     ENDLS2      ;                   ENDIF;
    0075 DB00        109 ELSE2    IN      DATA        ;               ELSE
    0077 E604        110          ANI     MINMSK      ;                   BEGIN
    0079 C2007F      111          JNZ     ENDIF05     ;
    007C C30082      112          JMP     CONT2       ;
    007F C3011A      113 ENDIF05  JMP     ENDIF9      ;
    0082 DB00        114 CONT2    IN      DATA        ;
    0084 E602        115          ANI     HRSMSK      ;
```

(130)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

0086 CA011A      116          JZ     ENDIF5      ;
0089 3A0003      117          LDA    MINFLG      ;
008C FE00        118          CPI    CLEAR       ;
008E C2009F      119          JNZ    ELSE6       ;
0091 210000      120          LXI    H,MINMSG    ;
0094 CD0000      121          CALL   TERMSG      ;
0097 3EFF        122          MVI    A,SET       ;
0099 320003      123          STA    MINFLG      ;
009C C30119      124 ENDIF6   JMP    ENDLS6      ;
009F 00          125 ELSE6    NOP                ;
00A0 3E00        126          MVI    A,00        ;
00A2 320000      127          STA    SECONDS     ;
00A5 3A0000      128          LDA    MINUTES     ;
00A8 C601        129          ADI    01          ;
00AA 320000      130          STA    MINUTES     ;
00AD FE3C        131          CPI    60          ;
00AF DA00CB      132          JC     ELSE7       ;
00B2 3E00        133          MVI    A,00        ;
00B4 320000      134          STA    MINUTES     ;
00B7 3EDF        135          MVI    A,DIGEN     ;
00B9 D300        136          OUT    PORTC       ;
00BB 3E00        137          MVI    A,ZERO      ;
00BD D300        138          OUT    VOICE       ;
00BF 3EFF        139          MVI    A,DISABL    ;
00C1 D300        140          OUT    PORTC       ;
00C3 3EFF        141          MVI    A,SET       ;
00C5 320000      142          STA    NEWMSG      ;
00C8 C30118      143 ENDIF7   JMP    ENDLS7      ;
00CB 3A0000      144 ELSE7    LDA    MINUTES     ;
00CE FE15        145          CPI    21          ;
00D0 D200E8      146          JNC    ELSE8       ;
00D3 3EDF        147          MVI    A,DIGEN     ;
00D5 D300        148          OUT    PORTC       ;
00D7 3A0000      149          LDA    MINUTES     ;
00DA D300        150          OUT    VOICE       ;
00DC 3EFF        151          MVI    A,DISABL    ;
00DE D300        152          OUT    PORTC       ;
00E0 3EFF        153          MVI    A,SET       ;
00E2 320000      154          STA    NEWMSG      ;
00E5 C30117      155 ENDIF8   JMP    ENDLS8      ;
00E8 00          156 ELSE8    NOP                ;
00E9 3A0000      157          LDA    MINUTES     ;
00EC 47          158          MOV    B,A         ;
00ED CD0000      159          CALL   BINBCD      ;
00F0 210000      160          LXI    H,DATSTK    ;
00F3 220000      161          SHLD   DATPTR      ;
00F6 C600        162          ADI    BASE        ;
00F8 77          163          MOV    M,A         ;
00F9 23          164          INX    H           ;
00FA 220000      165          SHLD   DATPTR      ;
00FD 78          166          MOV    A,B         ;
00FE FE00        167          CPI    00          ;
0100 C2010C      168          JNZ    ELSE9       ;
0103 3E00        169          MVI    A,SPACE20   ;
0105 2A0000      170          LHLD   DATPTR      ;
0108 77          171          MOV    M,A         ;
0109 C30111      172 ENDIF9   JMP    ENDLS9      ;
```

```
IF SETMIN = SET AND SETHRS = CLEAR THEN
   BEGIN

      IF MINFLG = CLEAR THEN
         BEGIN
            "GO TRANSFER MESSAGE"(,,,,MINMSG);

            MINFLG = SET
         ENDIF;
      ELSE
         BEGIN
            SECONDS = 00;


            MINUTES = MINUTES+1;

            IF MINUTES >= 60 THEN
               BEGIN
                  MINUTES = 00;


                  VOICE   = 'ZERO';


                  NEWMSG  = SET
               ENDIF;
            ELSE
               BEGIN
                  IF MINUTES <= 20 THEN
                     BEGIN

                        VOICE  = MINUTES;

                        NEWMSG = SET
                     ENDIF;
                  ELSE
                     BEGIN

                        "GO CONVERT BIN TO BCD"
                        (,MINUTES,,,)(TENS,UNITS,,,);
                        DATPTR = DATSTK;

                        ->DATPTR = TENS+BASE;

                        DATPTR = DATPTR+1;

                        IF UNITS = 00 THEN
                           BEGIN

                              ->DATPTR = SPACE20
                           ENDIF;
```

(131)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    010C 78           173 ELSE9    MOV    A,B          ;                                                    ELSE
    010D 2A0000       174          LHLD   DATPTR       ;                                                       BEGIN
    0110 77           175          MOV    M,A          ;                                                          ->DATPTR = UNITS
    0111 210000       176 ENDLS9   LXI    H,BCDMSG     ;                                                       ENDELSE;
    0114 CD0000       177          CALL   TFRMSG       ;                                                    "GO TRANSFER MESSAGE"
                      178                              ;                                                       (,,,,BCDMSG);
    0117 00           179 ENDLS8   NOP                 ;                                                 ENDELSE;
    0118 00           180 ENDLS7   NOP                 ;                                           ENDELSE;
    0119 00           181 ENDLS6   NOP                 ;                                     ENDELSE;
    011A 00           182 ENDIF5   NOP                 ;                                           ENDIF;
    011B 00           183 ENDLS2   NOP                 ;                                     ENDELSE;
    011C C30139       184 ENDIF1   JMP    END          ;                               ENDIF;
    011F DB00         185 ELSE1    IN     DATA         ;                          ELSE
    0121 E602         186          ANI    HRSMSK       ;                             BEGIN
    0123 CA012B       187          JZ     ENDIF10      ;                                 IF SETHRS = CLEAR THEN
    0126 3E00         188          MVI    A,CLEAR      ;                                    BEGIN
    0128 320002       189          STA    HRSFLG       ;                                       HRSFLG = CLEAR
    012B DB00         190 ENDIF10  IN     DATA         ;                                    ENDIF;
    012D E604         191          ANI    MINMSK ;     ;
    012F CA0137       192          JZ     ENDIF11      ;                                 IF MINFLG = CLEAR THEN
    0132 3E00         193          MVI    A,CLEAR      ;                                    BEGIN
    0134 320003       194          STA    MINFLG       ;                                       MINFLG = CLEAR
    0137 00           195 ENDIF11  NOP                 ;                                    ENDIF;
    0138 00           196 ENDLS1   NOP                 ;                               ENDELSE;
    0139 00           197 END      NOP                 ;  END;
    013A C9           198          RET                 ;RETURN;

Errors=    0
```

(132)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1 "8085"
                      3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                      4  EXPAND
                      5 ********************************************************************
                      6 *                                                                  *
                      7 * PRODUCT:        BLIND OPERATOR CONSOLE AID  FOR U.F.P.           *
                      8 *                                                                  *
                      9 * PRODUCT CODE:   9-XXX-XXXXX-XXX                                 *
                     10 *                                                                  *
                     11 * MODULE:         LAMPSCAN                                        *
                     12 *                                                                  *
                     13 * FUNCTION:       SCANS THE LAMPS VIA THE PIA AND PLACES THE      *
                     14 *                 BIT GIVING THE STATE OF THE LAMP IN THE LSB     *
                     15 *                 OF SEQUENTIAL BYTES OF LAMPSTACK. FIRST THE     *
                     16 *                 EXTENSIONS.EXCHANGE LINES,THEN LOOPS.           *
                     17 *                                                                  *
                     18 * CALLS:          NONE                                            *
                     19 *                                                                  *
                     20 * ON TAPE:        NOT YET                                         *
                     21 *                                                                  *
                     22 * LINKING:        LINK BLIND                                      *
                     23 * INSTRUCTIONS                                                     *
                     24 *                                                                  *
                     25 * FILE HISTORY:   STARTED 8/7/82                                  *
                     26 *                                                                  *
                     27 * 8085 CONVERSION STARTED 14/09/83                                *
                     28 *                                                                  *
                     29 * 0.0 08/07/82    INITIAL CONCEPT                                 *
                     30 *                                                                  *
                     31 *                                                                  *
                     32 ********************************************************************
                     33
                     34          GLOBAL              LMPSCN,LMPSTK,EXTSTK,EXCSTK,LOPSTK
                     35
                     36          EXTERNAL            PORTA,PORTB
                     37
                     38          DATA                ;VAR:
0000                 39 EXTSTK   DS      100         ;       LMPSTK : ARRAY OF LAMP ( 0-153 );
0064                 40 EXCSTK   DS      20          ;
0078                 41 LOPSTK   DS      34          ;
                <0000>42 LMPSTK  EQU     EXTSTK      ;
                     43                              ;
                     44                              ;       LAMP   : BYTE;
                     45                              ;       LMPNUM : BYTE ( 0,,,153 );
                     46                              ;       LMPADD : BYTE;
                     47                              ;       LMPSTT : BYTE;
                     48                              ;       DATA   : BYTE;
                     49                              ;
                     50                              ;CONST:
                <0001>51 LMPMSK  EQU     00000001B   ;       LMPMSK = 00000001B;
                <00FE>52 CLRMSK  EQU     11111110B   ;       CLRMSK = 11111110B;
                     53                              ;       @LMPNUM= ACCB;
                <0099>54 STKEND  EQU     153         ;       END OF LMPSTK;
                <0000>55 LMPADD  EQU     PORTA       ;       PORT A OF PIA;
                <0000>56 DATA    EQU     PORTB       ;       PORT B OF PIA;
                     57                              ;
                     58                              ;TYPE:
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                             59                              ;       LAMP    = RECORD OF BITS ( 0-7 )
                             60                              ;                   BIT0 - LMPSTAT CURR. SCAN
                             61                              ;                   BIT1 = LMPSTAT LAST  SCAN
                             62                              ;                   BIT2 = LAST TIMED STATE
                             63                              ;                   BIT3 = MESSAGE ON STACK
                             64                              ;                   BITS(4-7) - STATE TIMER
                             65                              ;       DATA    = RECORD OF BITS ( 0-7 )
                             66                              ;                   BIT0 = LAMP STATE
                             67                              ;                   BIT1 = SET HOURS
                             68                              ;                   BIT2 = SET MINUTES
                             69                              ;                   BITS(3-5) UNDEFINED
                             70                              ;                   BIT6 = GETMSG
                             71                              ;                   BIT7 = DIGRDY
                             72                              ;*************************************************
                             73                              ;
                             74          PROG                ;PROCEDURE "LMPSCN";
                             75                              ;   BEGIN
0000 210000                  76 LMPSCN   LXI     H,LMPSTK    ;       LMPPTR = LMPSTK;
0003 0600                    77          MVI     B,0         ;
0005 78                      78 LOOP     MOV     A,B         ;       FOR LMPNUM = 0 TO 153 DO
0006 FE9A                    79          CPI     STKEND+1    ;
0008 CA001C                  80          JZ      ENDFOR      ;           BEGIN
                             81                              ;
000B D300                    82          OUT     LMPADD      ;               LMPADD    = LMPNUM;
000D DB00                    83          IN      DATA        ;
000F E601                    84          ANI     LMPMSK      ;               LMPSTT    = DATA AND LMPMSK;
0011 57                      85          MOV     D,A         ;
0012 7E                      86          MOV     A,M         ;               CLEAR LAST STATE OF BIT0;
0013 E6FE                    87          ANI     CLRMSK      ;
0015 B2                      88          ORA     D           ;               LMPPTR = LMPPTR OR LMPSTT;
0016 77                      89          MOV     M,A         ;
0017 23                      90 NXTFOR   INX     H           ;               LMPPTR    = LMPPTR + 1
0018 04                      91          INR     B           ;               LMPNUM    = LMPNUM + 1
0019 C30005                  92          JMP     LOOP        ;
001C 00                      93 ENDFOR   NOP                 ;           ENDFOR;
001D 00                      94 END      NOP                 ;   END;
001E C9                      95          RET                 ;RETURN;

Errors=   0
```

(134)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                        1 "8085"
                        2  NAME  "9-XXX-XXXXX-XXX ISS 1"
                        3  EXPAND
                        4 ************************************************************
                        5 *                                                          *
                        6 * PRODUCT:        BLIND OPERATOR CONSOLE AID               *
                        7 *                                                          *
                        8 * PRODUCT CODE:   9-XXX-XXXXX-XXX                         *
                        9 *                                                          *
                       10 * MODULE:         LAMPSTATE                                *
                       11 *                                                          *
                       12 * FUNCTION:       CHECKS THE STATUS OF EACH SWITCHBOARD LAMP*
                       13 *                 ON THE CURRENT SCAN AND COMPARES IT WITH THE*
                       14 *                 STATUS ON THE PREVIOUS SCAN. THE STATUS I.E.*
                       15 *                 OFF,ON,OR TIMING TO OFF OR ON,IS DECIDED AND*
                       16 *                 BITS 0,1,2 REPRESENTING "NOW","WAS" & "STATE"*
                       17 *                 ARE UPDATED.                             *
                       18 *                                                          *
                       19 * CALLS:          NONE                                     *
                       20 *                                                          *
                       21 * ON TAPE:        NOT YET                                  *
                       22 *                                                          *
                       23 * LINKING:        LINK BLIND                               *
                       24 * INSTRUCTIONS                                             *
                       25 *                                                          *
                       26 * FILE HISTORY:   STARTED 8/7/82                          *
                       27 *                                                          *
                       28 * 8085 CONVERSION STARTED 12/09/83   HHHHHHHHHHHHHHHHHHHHHHHH*
                       29 *                                                          *
                       30 * 0.0 08/07/82    INITIAL CONCEPT                         *
                       31 *                                                          *
                       32 *                                                          *
                       33 ************************************************************
                       34
                       35         GLOBAL               LMPSTT
                       36
                       37         EXTERNAL             LMPSTK
                       38
                       39                              ;                LAMP BITS
                       40                              ; ***************************************
                       41                              ; * 7 * 6 * 5 * 4 * 3 * 2 * 1 * 0 *
                       42                              ; *  STATE TIMER  *MESS?*STATE* WAS * NOW *
                       43                              ; ***************************************
                       44                              ;
                       45         DATA                 ;VAR:
0000                   46 LMPPTR  DS      2            ;       LMPPTR : PTR;
0002                   47 LMPNUM  DS      1            ;       LMPNUM : BYTE;
0003                   48 TOGGLE  DS      1            ;       TOGGLE : BYTE;
0004                   49 FLAG    DS      1            ;       FLAG   : BYTE;
                       50                              ;
                       51                              ;CONST:
<0000>                 52 OFFOFF  EQU     00000000B    ;       OFFOFF = 00000000B;
<0001>                 53 OFFON   EQU     00000001B    ;       OFFON  = 00000001B;
<0002>                 54 ONOFF   EQU     00000010B    ;       ONOFF  = 00000010B;
<0003>                 55 ONON    EQU     00000011B    ;       ONON   = 00000011B;
<0003>                 56 SCNMSK  EQU     00000011B    ;       SCNMSK = 00000011B;
<0004>                 57 STTMSK  EQU     00000100B    ;       STTMSK = 00000100B;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                <00F0>    58 TIMMSK  EQU    11110000B  ;       TIMMSK = 11110000B;
                <00FC>    59 BTSMSK  EQU    11111100B  ;       BTSMSK = 11111100B;
                <0010>    60 TIMINC  EQU    00010000B  ;       TIMINC = 00010000B;
                <0020>    61 ONTIM   EQU    00100000B  ;       ONTIM  = 00100000B;
                <0020>    62 OFFTIM  EQU    00100000B  ;       OFFTIM = 00100000B;
                <00F0>    63 EXCOFF  EQU    11110000B  ;       EXCOFF = 11110000B;
                <000F>    64 ZERTIM  EQU    00001111B  ;       ZERTIM = 00001111B;
                <0008>    65 MESSON  EQU    00001000B  ;       MESSON = 00001000B;
                <00F7>    66 MESOFF  EQU    11110111B  ;       MESOFF = 11110111B;
                <00FB>    67 OFFSTT  EQU    11111011B  ;       OFFSTT = 11111011B;
                <0004>    68 ONSTT   EQU    00000100B  ;       ONSTT  = 00000100B;
                <0000>    69 CLEAR   EQU    00000000B  ;       CLEAR  = 00000000B;
                <00FF>    70 SET     EQU    11111111B  ;       SET    = 11111111B;
                          71                           ;*************************************
                          72                           ;
                          73         PROG               ;PROCEDURE "LMPSTT";
0000 3A0003               74 LMPSTT  LDA    TOGGLE     ;   BEGIN
0003 FEFF                 75         CPI    SET        ;
0005 CA0010               76         JZ     YESSET     ;
0008 3EFF                 77         MVI    A,SET      ;
000A 320003               78         STA    TOGGLE     ;
000D C30015               79         JMP    ENDTOG     ;
0010 3E00                 80 YESSET  MVI    A,CLEAR    ;
0012 320003               81         STA    TOGGLE     ;       TOGGLE = NOT TOGGLE;
0015 00                   82 ENDTOG  NOP                ;
0016 3E01                 83         MVI    A,01H      ;
0018 320002               84         STA    LMPNUM     ;
001B 3A0002               85 FORLOP  LDA    LMPNUM     ;
001E FE9B                 86         CPI    155        ;       FOR LMPNUM = 1 TO 154 DO
0020 D2012C               87         JNC    ENDFOR     ;
0023 3D                   88         DCR    A          ;           BEGIN
0024 4F                   89         MOV    C,A        ;
0025 0600                 90         MVI    B,00H      ;
0027 210000               91         LXI    H,LMPSTK   ;
002A 09                   92         DAD    B          ;
002B 220000               93         SHLD   LMPPTR     ;               LMPPTR = LMPSTK+(LMPNUM-1);
002E 7E                   94         MOV    A,M        ;               LMPBYT = ->LMPPTR;
002F E603                 95         ANI    SCNMSK     ;               LMPBYT = LMPBYT AND SCNMSK;
                          96                           ;               CASE LMPBYT OF
0031 FE00                 97         CPI    OFFOFF     ;
0033 C200B0               98         JNZ    NEXT1      ;               OFFOFF:
0036 7E                   99         MOV    A,M        ;                   BEGIN
0037 E604                100         ANI    STTMSK     ;
0039 CA00A3              101         JZ     ELSE1      ;                       IF STATE = ON THEN
003C 3E00                102         MVI    A,CLEAR    ;                           BEGIN
003E 320004              103         STA    FLAG       ;                               FLAG = CLEAR;
0041 3A0002              104         LDA    LMPNUM     ;
0044 FE65                105         CPI    101        ;
0046 DA0056              106         JC     TRUE1      ;
0049 FE79                107         CPI    121        ;
004B D20056              108         JNC    TRUE1      ;
004E 3A0003              109         LDA    TOGGLE     ;
0051 FEFF                110         CPI    SET        ;
0053 C2005A              111         JNZ    ENDIF7     ;                   IF LMPNUM <101 OR >=121 OR TOGGLE = SET THEN
0056 7E                  112 TRUE1  MOV    A,M        ;                       BEGIN
0057 C610                113         ADI    TIMINC     ;
0059 77                  114         MOV    M,A        ;                           TIMER = TIMER+1
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    00C8 E60F          172          ANI    ZERTIM      ;                    TIMER  = ZERO;
    00CA E6FC          173          ANI    BTSMSK      ;
    00CC F602          174          ORI    ONOFF       ;                    SCNBTS = ONOFF
    00CE 77            175          MOV    M,A         ;
    00CF 00            176 ENDLS3   NOP                 ;                ENDLSE;
    00D0 C30122        177          JMP    NXTFOR      ;            ENDOFFON;
                       178                              ;
    00D3 FE02          179 NEXT2    CPI    ONOFF       ;        ONOFF:
    00D5 C200F6        180          JNZ    NEXT3       ;
    00D8 7E            181          MOV    A,M         ;            BEGIN
    00D9 E604          182          ANI    STTMSK      ;               IF STATE = ON THEN
    00DB CA00E9        183          JZ     ELSE4       ;
    00DE 7E            184          MOV    A,M         ;                   BEGIN
    00DF E60F          185          ANI    ZERTIM      ;                       TIMER  = ZERO;
    00E1 E6FC          186          ANI    BTSMSK      ;
    00E3 F600          187          ORI    OFFOFF      ;                       SCNBTS = OFFOFF
    00E5 77            188          MOV    M,A         ;
    00E6 C300F2        189 ENDIF4   JMP    ENDLS4      ;                   ENDIF;
    00E9 00            190 ELSE4    NOP                 ;               ELSE
    00EA 7E            191          MOV    A,M         ;                   BEGIN
    00EB E60F          192          ANI    ZERTIM      ;                       TIMER  = ZERO;
    00ED E6FC          193          ANI    BTSMSK      ;
    00EF F600          194          ORI    OFFOFF      ;                       SCNBTS = OFFOFF
    00F1 77            195          MOV    M,A         ;
    00F2 00            196 ENDLS4   NOP                 ;                   ENDELSE;
    00F3 C30122        197          JMP    NXTFOR      ;            ENDONOFF;
                       198                              ;
    00F6 00            199 NEXT3    NOP                 ;        ONON:
    00F7 7E            200          MOV    A,M         ;            BEGIN
    00F8 E604          201          ANI    STTMSK      ;               IF STATE = OFF THEN
    00FA C20118        202          JNZ    ELSE5       ;
    00FD 7E            203          MOV    A,M         ;                   BEGIN
    00FE C610          204          ADI    TIMINC      ;
    0100 77            205          MOV    M,A         ;                       TIMER = TIMER +1;
    0101 E6F0          206          ANI    TIMMSK      ;
    0103 FE20          207          CPI    ONTIM       ;                       IF TIMER >= ONTIM THEN
    0105 DA0114        208          JC     ENDIF6      ;
    0108 7E            209          MOV    A,M         ;                           BEGIN
    0109 E60F          210          ANI    ZERTIM      ;                               TIMER  = ZERO;
    010B F604          211          ORI    ONSTT       ;                               STATE  = ON;
    010D E6FC          212          ANI    BTSMSK      ;
    010F F602          213          ORI    ONOFF       ;                               SCNBTS  = ONOFF;
    0111 E6F7          214          ANI    MESOFF      ;                               MESSAGE = OFF;
    0113 77            215          MOV    M,A         ;
    0114 00            216 ENDIF6   NOP                 ;                           ENDIF;
    0115 C30121        217 ENDIF5   JMP    ENDLS5      ;                   ENDIF;
    0118 00            218 ELSE5    NOP                 ;               ELSE
    0119 7E            219          MOV    A,M         ;                   BEGIN
    011A E60F          220          ANI    ZERTIM      ;                       TIMER  = ZERO;
    011C E6FC          221          ANI    BTSMSK      ;
    011E F602          222          ORI    ONOFF       ;                       SCNBTS = ONOFF
    0120 77            223          MOV    M,A         ;
    0121 00            224 ENDLS5   NOP                 ;                   ENDELSE;
                       225                              ;            ENDONON;
    0122 3A0002        226 NXTFOR   LDA    LMPNUM      ;
    0125 3C            227          INR    A           ;
    0126 320002        228          STA    LMPNUM      ;
```

(138)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0129 C30018      229         JMP    FORLOP      ;
    012C 00          230 ENDFOR  NOP                ;          ENDFOR;
    012D 00          231 END     NOP                ;    END;
    012E C9          232         RET                ;RETURN;

Errors=    0
```

(139)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                        1 "8085"
                        3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                        4  EXPAND
                        5 ********************************************************************
                        6 *                                                                  *
                        7 * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.            *
                        8 *                                                                  *
                        9 * PRODUCT CODE:   9-XXX-XXXXX-XXX                                 *
                       10 *                                                                  *
                       11 * MODULE:         SORTEXC                                        *
                       12 *                                                                  *
                       13 * FUNCTION:       SORTS OUT THE STATE OF THE EXCHANGE LAMPS       *
                       14 *                 AND GENERATES LAMP NUMBERS ON THE EXCHANGE      *
                       15 *                 MESSAGE STACK, READY FOR "GENMESSAGE" TO        *
                       16 *                 PRODUCE OUTPUT IN BUFFER. NUMBERS ARE PUT ON    *
                       17 *                 THE STACK IN A FIFO ORDER, AND ARE REMOVED      *
                       18 *                 IF THE LAMP GOES OUT. IF SO, THE STACK IS       *
                       19 *                 COLLAPSED.                                      *
                       20 *                                                                  *
                       21 * CALLS:          EXCMES                                         *
                       22 *                                                                  *
                       23 * ON TAPE:        NOT YET                                        *
                       24 *                                                                  *
                       25 * LINKING:        LINK BLIND                                     *
                       26 * INSTRUCTIONS                                                    *
                       27 *                                                                  *
                       28 * FILE HISTORY:   STARTED 8/7/82                                 *
                       29 *                                                                  *
                       30 * 8085 CONVERSION STARTED 20/09/83                               *
                       31 *                                                                  *
                       32 * 0.0 08/07/82    INITIAL CONCEPT                                *
                       33 *                                                                  *
                       34 *                                                                  *
                       35 ********************************************************************
                       36
                       37         GLOBAL                SRTEXC
                       38
                       39         EXTERNAL              EXCMES,EXCSTK
                       40
                       41                               ;
                       42                               ;CONST:
<0004>                 43 STATE   EQU     00000100B     ;       STATE   = 00000100B;
<0008>                 44 MESAGE  EQU     00001000B     ;       MESSAGE = 00001000B;
<0000>                 45 OFF     EQU     00000000B     ;       OFF     = 00000000B;
<00FF>                 46 ON      EQU     11111111B     ;       ON      = 11111111B;
                       47                               ;
                       48                               ;
                       49                               ;
                       50                               ;
                       51         DATA                  ;VAR:
                       52                               ;       EXCNUM  : BYTE;
                       53                               ;       LAMP    : BYTE;
0000                   54 TEMPB   DS      1             ;       TEMPB   : BYTE;
0001                   55 TEMPX   DS      2             ;       TEMPX   : PTR;
                       56                               ;
                       57         PROG                  ;PROCEDURE "SRTEXC"
                       58                               ;   BEGIN
```

(140)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0000 210000         59 SRTEXC LXI     H,EXCSTK    ;       PTR = EXCSTK;
    0003 0601           60        MVI     B,01H       ;
    0005 78             61 LOOP   MOV     A,B         ;
    0006 FE15           62        CPI     21          ;       FOR EXCNUM = 1 TO 20 DO
    0008 D2004C         63        JNC     ENDFOR      ;           BEGIN
    000B 7E             64        MOV     A,M         ;               LAMP = ->PTR;
    000C E604           65        ANI     STATE       ;               IF STATE = ON AND MESSAGE = OFF THEN
    000E CA002D         66        JZ      ELSE        ;
    0011 7E             67        MOV     A,M         ;
    0012 E608           68        ANI     MESAGE      ;
    0014 C2002A         69        JNZ     ENDIF       ;
    0017 78             70        MOV     A,B         ;                   BEGIN
    0018 320000         71        STA     TEMPB       ;
    001B 3EFF           72        MVI     A,ON        ;
    001D 220001         73        SHLD    TEMPX       ;
    0020 CD0000         74        CALL    EXCMES      ;                       "DO EXCHANGE MESSAGE"(ON,EXCNUM,,,)
    0023 3A0000         75        LDA     TEMPB       ;
    0026 47             76        MOV     B,A         ;
    0027 2A0001         77        LHLD    TEMPX       ;
    002A C30047         78 ENDIF  JMP     NXTFOR      ;                   ENDIF;
    002D 7E             79 ELSE   MOV     A,M         ;
    002E E608           80        ANI     MESAGE      ;               ELSE IF STATE = OFF AND MESSAGE = ON THEN
    0030 CA0046         81        JZ      ENDELS      ;
    0033 78             82        MOV     A,B         ;                   BEGIN
    0034 320000         83        STA     TEMPB       ;
    0037 3E00           84        MVI     A,OFF       ;
    0039 220001         85        SHLD    TEMPX       ;
    003C CD0000         86        CALL    EXCMES      ;                       "DO EXCHANGE MESSAGE"(OFF,EXTNUM,,,)
    003F 3A0000         87        LDA     TEMPB       ;
    0042 47             88        MOV     B,A         ;
    0043 2A0001         89        LHLD    TEMPX       ;
    0046 00             90 ENDELS NOP               ;                   ENDELSE;
    0047 04             91 NXTFOR INR     B           ;
    0048 23             92        INX     H           ;
    0049 C30005         93        JMP     LOOP        ;
    004C 00             94 ENDFOR NOP               ;           ENDFOR;
    004D 00             95 END    NOP               ;       END;
    004E C9             96        RET               ;RETURN;

Errors=    0
```

(141)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                              1 "8085"
                              3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                              4  EXPAND
                              5 ************************************************************
                              6 *                                                          *
                              7 * PRODUCT:          BLIND OPERATOR CONSOLE AID FOR U.F.P.  *
                              8 *                                                          *
                              9 * PRODUCT CODE:     9-XXX-XXXXX-XXX                        *
                             10 *                                                          *
                             11 * MODULE:           SORTEXT                                *
                             12 *                                                          *
                             13 * FUNCTION:         SORTS OUT THE STATE OF THE EXTENSION LAMPS *
                             14 *                   AND GENERATES LAMP NUMBERS ON THE EXTENSION *
                             15 *                   MESSAGE STACK, READY FOR "GENMESSAGE" TO *
                             16 *                   PRODUCE OUTPUT IN BUFFER. NUMBERS ARE PUT ON *
                             17 *                   THE STACK IN A FIFO ORDER, AND ARE REMOVED *
                             18 *                   IF THE LAMP GOES OUT. IF SO, THE STACK IS *
                             19 *                   COLLAPSED.                             *
                             20 *                                                          *
                             21 * CALLS:            EXTMES                                 *
                             22 *                                                          *
                             23 * ON TAPE:          NOT YET                                *
                             24 *                                                          *
                             25 * LINKING:          LINK BLIND                             *
                             26 * INSTRUCTIONS                                             *
                             27 *                                                          *
                             28 * FILE HISTORY:     STARTED 8/7/82                         *
                             29 *                                                          *
                             30 * 8185 CONVERSION STARTED 13/09/83                        *
                             31 *                                                          *
                             32 * 0.0 03/07/82      INITIAL CONCEPT                        *
                             33 *                                                          *
                             34 *                                                          *
                             35 ************************************************************
                             36
                             37          GLOBAL              SRTEXT
                             38
                             39          EXTERNAL            EXTMES;EXTSTK
                             40
                             41                              ;
                             42                              ;CONST:
<0004>                       43 STATE    EQU     00000100B   ;       STATE   = 00000100B;
<0008>                       44 MESAGE   EQU     00001000B   ;       MESSAGE = 00001000B;
<0000>                       45 OFF      EQU     00000000B   ;       OFF     = 00000000B;
<00FF>                       46 ON       EQU     11111111B   ;       ON      = 11111111B;
                             47                              ;
                             48                              ;
                             49                              ;
                             50                              ;
                             51          DATA                ;VAR:
                             52                              ;       EXTNUM  : BYTE;
                             53                              ;       LAMP    : BYTE;
0000                         54 TEMPB    DS      1           ;       TEMPB   : BYTE;
0001                         55 TEMPX    DS      2           ;       TEMPX   : PTR;
                             56                              ;
                             57          PROG                ;PROCEDURE "SRTEXT"
                             58                              ;  BEGIN
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0000 210000        59 SRTEXT LXI     H,EXTSTK    ;       PTR = EXTSTK;
    0003 0601          60        MVI     B,01H       ;
    0005 78            61 LOOP   MOV     A,B         ;
    0006 FE65          62        CPI     101         ;       FOR EXTNUM = 1 TO 100 DO
    0008 D2004C        63        JNC     ENDFOR      ;           BEGIN
    000B 7E            64        MOV     A,M         ;               LAMP = ->PTR;
    000C E604          65        ANI     STATE       ;               IF STATE = ON AND MESSAGE = OFF THEN
    000E CA002D        66        JZ      ELSE        ;
    0011 7E            67        MOV     A,M         ;
    0012 E608          68        ANI     MESAGE      ;
    0014 C2002A        69        JNZ     ENDIF       ;
    0017 78            70        MOV     A,B         ;                   BEGIN
    0018 320000        71        STA     TEMPB       ;
    001B 3EFF          72        MVI     A,ON        ;
    001D 220001        73        SHLD    TEMPX       ;
    0020 CD0000        74        CALL    EXTMES      ;                       "DO EXTENSION MESSAGE"(ON,EXTNUM,,,)
    0023 3A0000        75        LDA     TEMPB       ;
    0026 47            76        MOV     B,A         ;
    0027 2A0001        77        LHLD    TEMPX       ;
    002A C30047        78 ENDIF  JMP     NXTFOR      ;                   ENDIF;
    002D 7E            79 ELSE   MOV     A,M         ;
    002E E608          80        ANI     MESAGE      ;               ELSE IF STATE = OFF AND MESSAGE = ON THEN
    0030 CA0046        81        JZ      ENDELS      ;
    0033 78            82        MOV     A,B         ;                   BEGIN
    0034 320000        83        STA     TEMPB       ;
    0037 3E00          84        MVI     A,OFF       ;
    0039 220001        85        SHLD    TEMPX       ;
    003C CD0000        86        CALL    EXTMES      ;                       "DO EXTENSION MESSAGE"(OFF,EXTNUM,,,)
    003F 3A0000        87        LDA     TEMPB       ;
    0042 47            88        MOV     B,A         ;
    0043 2A0001        89        LHLD    TEMPX       ;
    0046 00            90 ENDELS NOP               ;                   ENDELSE;
    0047 04            91 NXTFOR INR     B           ;
    0048 23            92        INX     H           ;
    0049 C30005        93        JMP     LOOP        ;
    004C 00            94 ENDFOR NOP               ;           ENDFOR;
    004D 00            95 END    NOP               ;       END;
    004E C9            96        RET               ;RETURN;

Errors=    0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                        1 "8085"
                        3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                        4  EXPAND
                        5 ;*****************************************************************
                        6 ;*                                                               *
                        7 ;* PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.        *
                        8 ;*                                                               *
                        9 ;* PRODUCT CODE:   9-XXX-XXXXX-XXX                              *
                       10 ;*                                                               *
                       11 ;* MODULE:         LOOPSTATE                                    *
                       12 ;*                                                               *
                       13 ;* FUNCTION:       SORTS OUT THE STATE OF THE LOOPS USING THE    *
                       14 ;*                  TWO SEQUENTIAL BYTES FOR "ANSWER" AND "CALL" *
                       15 ;*                  LAMPS ON THE SWITCHBOARD. A "LOOPSTATE" NO.   *
                       16 ;*                  IS GENERATED AND PUT IN BITS 0 & 1 OF THE     *
                       17 ;*                  THE FIRST BYTE OF EACH SET OF 5 LOOP STATUS    *
                       18 ;*                  BYTES. THE STATE ON THE PREVIOUS 100MS SCAN    *
                       19 ;*                  IS SHIFTED TO BITS 2 & 3. E.G. "????XX10" TO   *
                       20 ;*                  "????1011" IF "11" THIS SCAN.                 *
                       21 ;*                                                               *
                       22 ;* CALLS:          NONE                                         *
                       23 ;*                                                               *
                       24 ;* ON TAPE:        NOT YET                                      *
                       25 ;*                                                               *
                       26 ;* LINKING:        LINK BLIND                                   *
                       27 ;* INSTRUCTIONS                                                 *
                       28 ;*                                                               *
                       29 ;* FILE HISTORY:   STARTED 8/7/82                               *
                       30 ;*                                                               *
                       31 ;*                                                               *
                       32 ;* 0.0 08/07/82    INITIAL CONCEPT                              *
                       33 ;*                                                               *
                       34 ;*****************************************************************
                       35
                       36          GLOBAL              LOPSTT,STTSTK
                       37
                       38          EXTERNAL            LOPSTK
                       39
                       40                              ;
                       41                              ;CONST:
<0004>                 42 STATE    EQU     00001000B   ;       STATE   = 00001000B;
<000C>                 43 WASMSK   EQU     00001100B   ;       WASMSK  = 00001100B;
<00F0>                 44 STTMSK   EQU     11110000B   ;       STTMSK  = 11110000B;
                       45                              ;
                       46                              ;TYPE:
<0005>                 47 LOOP     EQU     5           ;       LOOP    : RECORD OF BYTE (1,,5);
                       48                              ;
                       49                              ;               BYTE5 = *               *
                       50                              ;                       *****************
                       51                              ;               BYTE4 = *               *
                       52                              ;                       *****************
                       53                              ;               BYTE3 = *               *
                       54                              ;                       *****************
                       55                              ;               BYTE2 = *               *
                       56                              ;                       *****************
                       57                              ;               BYTE1 = * GENERAL STATUS * WAS * NOW *
                       58                              ;                       *****************
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      59                                ;                    BITS   7 6    5 4    3 2    1 0
                      60                                ;
                      61          DATA                  ;VAR:
                      62                                ;       LAMP    : BYTE;
0000                  63 LOPNUM   DS      1             ;       LOPNUM  : BYTE;
0001                  64 STTSTK   DS      17*LOOP       ;       STTSTK  : ARRAY OF LOOP (1,,17);
                      65                                ;
                      66          PROG                  ;PROCEDURE "LOPSTT"
                      67                                ;   BEGIN
0000 3E01             68 LOPSTT   MVI     A,01H         ;
0002 320000           69          STA     LOPNUM        ;
0005 3A0000           70 FORLOOP  LDA     LOPNUM        ;
0008 FE12             71          CPI     18            ;       FOR LOPNUM = 1 TO 17 DO
000A D20046           72          JNC     ENDFOR        ;           BEGIN
000D 3D               73          DCR     A             ;
000E 47               74          MOV     B,A           ;
000F 87               75          ADD     A             ;
0010 87               76          ADD     A             ;
0011 80               77          ADD     B             ;
0012 4F               78          MOV     C,A           ;
0013 0600             79          MVI     B,00H         ;
0015 210001           80          LXI     H,STTSTK      ;
0018 09               81          DAD     B             ;               STTPTR = STTSTK+(5*(LOPNUM-1));
0019 3A0000           82          LDA     LOPNUM        ;
001C 3D               83          DCR     A             ;
001D 87               84          ADD     A             ;
001E 4F               85          MOV     C,A           ;
001F 110000           86          LXI     D,LOPSTK      ;
0022 EB               87          XCHG                  ;
0023 09               88          DAD     B             ;               LAMPTR = LOPSTK+(2*(LOPNUM-1));
0024 7E               89          MOV     A,M           ;               LAMP1  = ->LAMPTR;
0025 E604             90          ANI     STATE         ;               LAMP1  = LAMP1 AND STATE;
0027 0F               91          RRC                   ;               LAMP1  = LAMP1 ROR 1;
0028 47               92          MOV     B,A           ;
0029 23               93          INX     H             ;               LAMPTR = LAMPTR+1;
002A 7E               94          MOV     A,M           ;               LAMP2  = ->LAMPTR;
002B E604             95          ANI     STATE         ;               LAMP2  = LAMP2 AND STATE;
002D B0               96          ORA     B             ;               NOWSTT = LAMP1 OR LAMP2;
002E 0F               97          RRC                   ;               NOWSTT = NOWSTT ROR 1;
002F 47               98          MOV     B,A           ;
0030 1A               99          LDAX    D             ;               WASSTT = ->STTPTR;
0031 07              100          RLC                   ;               WASSTT = WASSTT ROL 2
0032 07              101          RLC                   ;
0033 E60C            102          ANI     WASMSK        ;               WASSTT = WASSTT AND WASMSK;
0035 B0              103          ORA     B             ;               STATUS = WASSTT OR NOWSTT;
0036 47              104          MOV     B,A           ;
0037 1A              105          LDAX    D             ;               BYTE1  = ->STTPTR;
0038 E6F0            106          ANI     STTMSK        ;               BYTE1  = BYTE1 AND STTMSK;
003A B0              107          ORA     B             ;               BYTE1  = BYTE1 OR STATUS;
003B 12              108          STAX    D             ;               ->STTPTR = BYTE1
003C 3A0000          109          LDA     LOPNUM        ;
003F 3C              110          INR     A             ;
0040 320000          111          STA     LOPNUM        ;
0043 C30005          112          JMP     FORLOOP       ;
0046 00              113 ENDFOR  NOP                    ;           ENDFOR;
0047 00              114 END     NOP                    ;       END;
0048 C9              115          RET                   ;RETURN;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                              1 "8085"
                              3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                              4  EXPAND
                              5 ;*****************************************************************
                              6 ;*                                                               *
                              7 ;* PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.        *
                              8 ;*                                                               *
                              9 ;* PRODUCT CODE:   9-XXX-XXXXX-XXX                              *
                             10 ;*                                                               *
                             11 ;* MODULE:         SRTLOP                                       *
                             12 ;*                                                               *
                             13 ;* FUNCTION:       THIS MODULE USES THE STATE OF THE LOOP AS     *
                             14 ;*                  SORTED OUT BY "LOPSTT" TO DETERMINE WHICH OF *
                             15 ;*                  16 POSSIBLE STATES AND HENCE ACTION TO TAKE  *
                             16 ;*                  FOR EACH OF 17 LOOPS. E.G. WAS "OFFON", IS   *
                             17 ;*                  NOW 'ONON" IS TAKEN TO BE "01 11". A STATE   *
                             18 ;*                  TABLE OF ADDRESSES OF FUNCTIONS, OFFSET BY THE *
                             19 ;*                  STATE NUMBER, IS USED TO FIND THE DESTINATION. *
                             20 ;*                                                               *
                             21 ;* CALLS:          STATES (AS E.G STATE1,,,,STATE5)             *
                             22 ;*                                                               *
                             23 ;* ON TAPE:        NOT YET                                      *
                             24 ;*                                                               *
                             25 ;* LINKING:        LINK BLIND                                   *
                             26 ;* INSTRUCTIONS                                                  *
                             27 ;*                                                               *
                             28 ;* FILE HISTORY:   STARTED 8/7/82                               *
                             29 ;*                                                               *
                             30 ;* 0.0 08/07/82    INITIAL CONCEPT                              *
                             31 ;*                                                               *
                             32 ;*                                                               *
                             33 ;*****************************************************************
                             34
                             35          GLOBAL              SRTLOP
                             36
                             37          EXTERNAL            STTSTK,STATE1,STATE2,STATE3,STATE4,STATE5
                             38
                             39                              ;CONST:
      <000F>                 40 STTMSK   EQU     00001111B   ;       STTMSK  = 00001111B;
                             41                              ;
                             42          DATA                ;VAR:
0000                         43 LOPNUM   DS      1           ;       LOPNUM  : BYTE;
0001                         44 STATPTR  DS      2           ;       STATPTR : PTR;
                             45                              ;
                             46                              ;
                             47                              ;
                             48                              ;
                             49                              ;TYPE:
                             50                              ;
                             51          PROC                ;
                             52                              ;
0000 0000                    53 STATBL   DW      STATE1      ;       STATBL  : RECORD OF ADDRESSES (1,,16)
0002 0000                    54          DW      STATE2      ;
0004 0000                    55          DW      STATE2      ;
0006 0000                    56          DW      STATE3      ;
0008 0000                    57          DW      STATE3      ;
000A 0000                    58          DW      STATE4      ;
```

(146)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    000C 0000          59          DW      STATE2          ;
    000E 0000          60          DW      STATE3          ;
    0010 0000          61          DW      STATE3          ;
    0012 0000          62          DW      STATE2          ;
    0014 0000          63          DW      STATE4          ;
    0016 0000          64          DW      STATE3          ;
    0018 0000          65          DW      STATE3          ;
    001A 0000          66          DW      STATE2          ;
    001C 0000          67          DW      STATE2          ;
    001E 0000          68          DW      STATE5          ;
                       69                                  ;
                       70                                  ;PROCEDURE "SRTLOP";
                       71                                  ;  BEGIN
    0020 3E01          72 SRTLOP   MVI     A,01H           ;
    0022 320000        73          STA     LOPNUM          ;
    0025 3A0000        74 FORLOOP  LDA     LOPNUM          ;
    0028 FE12          75          CPI     18              ;       FOR LOPNUM = 1 TO 17 DO
    002A D2005D        76          JNC     ENDFOR          ;           BEGIN
    002D 3D            77          DCR     A               ;
    002E 47            78          MOV     B,A             ;
    002F 87            79          ADD     A               ;
    0030 87            80          ADD     A               ;
    0031 80            81          ADD     B               ;
    0032 4F            82          MOV     C,A             ;
    0033 0600          83          MVI     B,00H           ;
    0035 210000        84          LXI     H,STTSTK        ;
    0038 09            85          DAD     B               ;
    0039 220001        86          SHLD    STATPTR         ;               STATPTR = STTSTK+(5*(LOPNUM-1));
    003C 7E            87          MOV     A,M             ;
    003D E60F          88          ANI     STTMSK          ;               STATE   = ->STATPTR AND STTMSK;
    003F 87            89          ADD     A               ;
    0040 4F            90          MOV     C,A             ;
    0041 210000        91          LXI     H,STATBL        ;
    0044 09            92          DAD     B               ;               JMPPTR  = STATBL+(2*STATE);
    0045 5E            93          MOV     E,M             ;
    0046 23            94          INX     H               ;
    0047 56            95          MOV     D,M             ;
    0048 2A0001        96          LHLD    STATPTR         ;
    004B EB            97          XCHG                    ;
    004C 3A0000        98          LDA     LOPNUM          ;
    004F 47            99          MOV     B,A             ;
    0050 CD0060       100          CALL    PROCED          ;               "DO PROCEDURE ->JMPPTR"(,LOPNUM,,STATPTR,);
    0053 3A0000       101 NXTFOR    LDA     LOPNUM          ;
    0056 3C           102          INR     A               ;
    0057 320000       103          STA     LOPNUM          ;
    005A C30025       104          JMP     FORLOOP         ;
    005D 00           105 ENDFOR    NOP                     ;           ENDFOR;
    005E 00           106 END       NOP                     ;   END;
    005F C9           107          RET                     ;RETURN;
                      108                                  ;
    0060 E9           109 PROCED    PCHL                    ; * ACTUALLY DO THE CALL TO POINTED PROCEDURE *

Errors=    0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1 "8085"
                      3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                      4  EXPAND
                      5 ;*****************************************************************
                      6 ;*                                                               *
                      7 ;* PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.        *
                      8 ;*                                                               *
                      9 ;* PRODUCT CODE:   9-XXX-XXXXX-XXX                              *
                     10 ;*                                                               *
                     11 ;* MODULE:         STATES                                       *
                     12 ;*                                                               *
                     13 ;* FUNCTION:       THIS MODULE DEFINES THE VARIOUS STATES THAT   *
                     14 ;*                  A LOOP CAN BE IN. THE PROCEDURE "SRTLOP" HAS  *
                     15 ;*                  PRODUCED THE POINTER TO THE SUB-PROCEDURE IN  *
                     16 ;*                  THIS MODULE. THE STATUS BLOCK OF 5 BYTES FOR  *
                     17 ;*                  A PARTICULAR LOOP IS POINTED TO BY INDEX PAIR *
                     18 ;*                  D,E AND IS PICTURED BELOW.                   *
                     19 ;*                                                               *
                     20 ;*                     ***************************************   *
                     21 ;*                  BYTE5 *       MAXIMUM WAIT TIME         *   *
                     22 ;*                     ***************************************   *
                     23 ;*                  BYTE4 *       NUMBER  OF  FLASHES       *   *
                     24 ;*                     ***************************************   *
                     25 ;*                  BYTE3 *            FLASH TIMER          *   *
                     26 ;*                     ***************************************   *
                     27 ;*                  BYTE2 *  ON/OFF/WAIT TIMER, MAX. 25,5 SEC *  *
                     28 ;*                     ***************************************   *
                     29 ;*                  BYTE1 * WAIT * FLSH * ON * OFF * WAS * NOW *  *
                     30 ;*                     ***************************************   *
                     31 ;*                  BITS#    7      6      5    4    3 2    1 0   *
                     32 ;*                                                               *
                     33 ;* CALLS:          NONE                                         *
                     34 ;*                                                               *
                     35 ;* ON TAPE:        NOT YET                                      *
                     36 ;*                                                               *
                     37 ;* LINKING:        LINK BLIND                                   *
                     38 ;* INSTRUCTIONS                                                  *
                     39 ;*                                                               *
                     40 ;* FILE HISTORY:   STARTED 8/7/82                               *
                     41 ;*                                                               *
                     42 ;* 0.0 08/07/82    INITIAL CONCEPT                              *
                     43 ;*                                                               *
                     44 ;*                                                               *
                     45 ;*****************************************************************
                     46
                     47          GLOBAL              STATE1,STATE2,STATE3,STATE4,STATE5
                     48          GLOBAL              OFFLG
                     49
                     50          EXTERNAL            CANCEL,CHKLOP,LOPMES
                     51
                     52                              ;CONST:
<00FF>               53 ON       EQU     0FFH        ;  ON      = 0FFH;
<0000>               54 OFF      EQU     000H        ;  OFF     = 000H;
<00FF>               55 SET      EQU     0FFH        ;  SET     = 0FFH;
<0000>               56 CLEAR    EQU     000H        ;  CLEAR   = 000H;
<000F>               57 STTMSK   EQU     00001111B   ;  STTMSK  = 00001111B;
<0080>               58 WAITFLG  EQU     10000000B   ;  WAITFLG = 10000000B;
```

(148)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

            <0040>     59 FLSHFLG EQU    01000000B   ;   FLSHFLG = 01000000B;
            <0020>     60 ONFLG   EQU    00100000B   ;   ONFLG   = 00100000B;
            <0010>     61 OFFLG   EQU    00010000B   ;   OFFLG   = 00010000B;
            <0014>     62 OFFLIM  EQU    20          ;   OFFLIM  = 20 (2,0 SEC.);
            <0014>     63 ONLIM   EQU    20          ;   ONLIM   = 20 (2,0 SEC.);
            <0004>     64 FLSHLIM EQU    4           ;   FLSHLIM =  4 (0,4 SEC.);
                       65                            ;
                       66         DATA               ;VAR:
0000                   67 LOPNUM  DS     1           ;   LOPNUM  : BYTE;
0001                   68 STATPTR DS     2           ;   STATPTR : PTR;
                       69                            ;
                       70                            ;TYPE:
                       71                            ;
                       72                            ;
                       73         PROG               ;PROCEDURE "STATE1";
                       74                            ;   BEGIN
0000 78                75 STATE1  MOV    A,B         ;
0001 320000            76         STA    LOPNUM      ;
0004 EB                77         XCHG               ;
0005 220001            78         SHLD   STATPTR     ;
0008 7E                79         MOV    A,M         ;
0009 E610              80         ANI    OFFLG       ;
000B C2005F            81         JNZ    ENDIF11     ;       IF OFFLG = CLEAR THEN
000E 7E                82         MOV    A,M         ;         BEGIN
000F E640              83         ANI    FLSHFLG     ;
0011 C20029            84         JNZ    ENDIF12     ;             IF FLSHFLG = CLEAR THEN
0014 23                85         INX    H           ;               BEGIN
0015 23                86         INX    H           ;
0016 34                87         INR    M           ;                   FLSHTIM = FLSHTIM+1;
0017 7E                88         MOV    A,M         ;
0018 FE04              89         CPI    FLSHLIM     ;                   IF FLSHTIM >= FLSHLIM THEN
001A DA0028            90         JC     ENDIF13     ;                     BEGIN
001D 23                91         INX    H           ;
001E 34                92         INR    M           ;                       FLSHNUM = FLSHNUM+1;
001F 2A0001            93         LHLD   STATPTR     ;
0022 7E                94         MOV    A,M         ;                       WAITFLG = CLEAR;
0023 E60F              95         ANI    STTMSK      ;                       FLSHFLG = SET;
0025 F640              96         ORI    FLSHFLG     ;                       ONFLG   = CLEAR;
0027 77                97         MOV    M,A         ;                       OFFLG   = CLEAR
0028 00                98 ENDIF13 NOP                ;                     ENDIF;
0029 00                99 ENDIF12 NOP                ;               ENDIF;
002A 2A0001           100         LHLD   STATPTR     ;
002D 23               101         INX    H           ;
002E 34               102         INR    M           ;             OFFTIM  = OFFTIM+1;
002F 7E               103         MOV    A,M         ;
0030 FE14             104         CPI    OFFLIM      ;
0032 DA005E           105         JC     ENDIF14     ;             IF OFFTIM >= OFFLIM THEN
0035 2A0001           106         LHLD   STATPTR     ;               BEGIN
0038 7E               107         MOV    A,M         ;                   WAITFLG = CLEAR;
0039 E60F             108         ANI    STTMSK      ;                   FLSHFLG = CLEAR;
003B F610             109         ORI    OFFLG       ;                   ONFLG   = CLEAR;
003D 77               110         MOV    M,A         ;                   OFFLG   = SET;
003E 23               111         INX    H           ;
003F 3600             112         MVI    M,CLEAR     ;                   OFFTIM  = CLEAR;
0041 23               113         INX    H           ;
0042 3600             114         MVI    M,CLEAR     ;                   FLSHTIM = CLEAR;
0044 23               115         INX    H           ;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0045 3600          116          MVI    M,CLEAR     ;                       FLSHNUM = CLEAR;
    0047 23            117          INX    H           ;
    0048 3600          118          MVI    M,CLEAR     ;                       WAITLIM = CLEAR;
    004A 3A0000        119          LDA    LOPNUM      ;
    004D 47            120          MOV    B,A         ;
    004E 3E00          121          MVI    A,OFF       ;
    0050 CD0000        122          CALL   LOPMES      ;                       "DO LOOP MESSAGE"(OFF,LOPNUM,,,);
    0053 3A0000        123          LDA    LOPNUM      ;
    0056 47            124          MOV    B,A         ;
    0057 3EFF          125          MVI    A,ON        ;
    0059 0E00          126          MVI    C,CANCEL    ;
    005B CD0000        127          CALL   LOPMES      ;                       "DO LOOP MESSAGE"(ON,LOPNUM,CANCEL,,)
    005E 00            128 ENDIF14  NOP                ;                   ENDIF;
    005F 00            129 ENDIF11  NOP                ;               ENDIF;
    0060 00            130 END1     NOP                ;   END;
    0061 C9            131          RET                ;RETURN;
                       132                             ;
                       133 ********************************************************************************
                       134                             ;
                       135                             ;PROCEDURE"STATE2";
                       136                             ;   BEGIN
    0062 78            137 STATE2   MOV    A,B        ;
    0063 320000        138          STA    LOPNUM      ;
    0066 EB            139          XCHG               ;
    0067 220001        140          SHLD   STATPTR     ;
    006A 7E            141          MOV    A,M         ;
    006B 23            142          INX    H           ;
    006C 23            143          INX    H           ;
    006D 23            144          INX    H           ;
    006E 23            145          INX    H           ;
    006F E610          146          ANI    OFFLG       ;       IF OFFLG = SET THEN
    0071 CA0079        147          JZ     ELSE21      ;           BEGIN
    0074 36FF          148          MVI    M,255       ;               WAITLIM = 255          * 25,5 SECS. *
    0076 C3007D        149 ENDIF21  JMP    ENDLS21     ;           ENDIF;
    0079 00            150 ELSE21   NOP                ;       ELSE
    007A 00            151          NOP                ;           BEGIN
    007B 3696          152          MVI    M,150       ;               WAITLIM = 150          * 15,0 SECS. *
    007D 00            153 ENDLS21  NOP                ;           ENDELSE;
    007E 2A0001        154          LHLD   STATPTR     ;       WAITFLG = CLEAR;
    0081 7E            155          MOV    A,M         ;       FLSHFLG = CLEAR;
    0082 E60F          156          ANI    STTMSK      ;       ONFLG   = CLEAR;
    0084 77            157          MOV    M,A         ;       OFFLG   = CLEAR;
    0085 23            158          INX    H           ;
    0086 3600          159          MVI    M,CLEAR     ;       WAITIM  = CLEAR;
    0088 23            160          INX    H           ;
    0089 3600          161          MVI    M,CLEAR     ;       FLSHTIM = CLEAR;
    008B 3A0000        162          LDA    LOPNUM      ;
    008E 47            163          MOV    B,A         ;
    008F 3E00          164          MVI    A,OFF       ;
    0091 CD0000        165          CALL   LOPMES      ;           "DO LOOP MESSAGE"(OFF,LOPNUM,,,)
    0094 00            166 END2     NOP                ;   END;
    0095 C9            167          RET                ;RETURN;
                       168                             ;
                       169 ********************************************************************************
                       170                             ;
                       171                             ;PROCEDURE"STATE3";
                       172                             ;   BEGIN
```

(150)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

0096 78           173 STATE3  MOV   A,B          ;
0097 320000       174         STA   LOPNUM       ;
009A EB           175         XCHG               ;
009B 220001       176         SHLD  STATPTR      ;
009E 7E           177         MOV   A,M          ;       WAITFLG = CLEAR;
009F E60F         178         ANI   STTMSK       ;       FLSHFLG = CLEAR;
00A1 77           179         MOV   M,A          ;       ONFLG   = CLEAR;
00A2 00           180         NOP                ;       OFFLG   = CLEAR;
00A3 23           181         INX   H            ;
00A4 3600         182         MVI   M,CLEAR      ;       WAITIM  = CLEAR;
00A6 23           183         INX   H            ;
00A7 3600         184         MVI   M,CLEAR      ;       FLSHTIM = CLEAR;
00A9 23           185         INX   H            ;
00AA 23           186         INX   H            ;
00AB 3600         187         MVI   M,CLEAR      ;       WAITLIM = CLEAR;
00AD 3A0000       188         LDA   LOPNUM       ;
00B0 47           189         MOV   B,A          ;
00B1 3E00         190         MVI   A,OFF        ;
00B3 CD0000       191         CALL  LOPMES       ;       "DO LOOP MESSAGE"(OFF,LOPNUM,,,)
00B6 00           192 END3    NOP                ;   END;
00B7 C9           193         RET                ;RETURN;
                  194                            ;
                  195 ***************************************************************************
                  196                            ;
                  197                            ;PROCEDURE"STATE4";
                  198                            ;   BEGIN
00B8 78           199 STATE4  MOV   A,B          ;
00B9 320000       200         STA   LOPNUM       ;
00BC EB           201         XCHG               ;
00BD 220001       202         SHLD  STATPTR      ;
00C0 7E           203         MOV   A,M          ;
00C1 E680         204         ANI   WAITFLG      ;
00C3 C20153       205         JNZ   ENDIF41      ;       IF WAITFLG = CLEAR THEN
00C6 7E           206         MOV   A,M          ;           BEGIN
00C7 E640         207         ANI   FLSHFLG      ;               IF FLSHFLG = CLEAR THEN
00C9 C20113       208         JNZ   ENDIF42      ;                   BEGIN
00CC 23           209         INX   H            ;
00CD 23           210         INX   H            ;
00CE 34           211         INR   M            ;                       FLSHTIM = FLSHTIM+1;
00CF 7E           212         MOV   A,M          ;
00D0 FE04         213         CPI   FLSHLIM      ;                       IF FLSHTIM >= FLSHLIM THEN
00D2 DA0112       214         JC    ENDIF43      ;                           BEGIN
00D5 2A0001       215         LHLD  STATPTR      ;
00D8 7E           216         MOV   A,M          ;                               OFFLG   = CLEAR;
00D9 E60F         217         ANI   STTMSK       ;                               ONFLG   = CLEAR;
00DB F640         218         ORI   FLSHFLG      ;                               FLSHFLG = SET;
00DD 77           219         MOV   M,A          ;                               WAITFLG = CLEAR;
00DE 23           220         INX   H            ;
00DF 23           221         INX   H            ;
00E0 23           222         INX   H            ;
00E1 34           223         INR   M            ;                               FLSHNUM = FLSHNUM+1;
00E2 7E           224         MOV   A,M          ;
00E3 FE04         225         CPI   4            ;                               IF FLSHNUM >= 4 THEN
00E5 DA0111       226         JC    ENDIF44      ;                                   BEGIN
00E8 2A0001       227         LHLD  STATPTR      ;
00EB 7E           228         MOV   A,M          ;                                       OFFLG   = CLEAR;
00EC E60F         229         ANI   STTMSK       ;                                       ONFLG   = CLEAR;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    00EE F680        230          ORI    WAITFLG     ;                    FLSHFLG = CLEAR;
    00F0 77          231          MOV    M,A         ;                    WAITFLG = SET;
    00F1 23          232          INX    H           ;
    00F2 3600        233          MVI    M,CLEAR     ;                    OFFTIM  = CLEAR;
    00F4 23          234          INX    H           ;
    00F5 3600        235          MVI    M,CLEAR     ;                    FLSHTIM = CLEAR;
    00F7 23          236          INX    H           ;
    00F8 3600        237          MVI    M,CLEAR     ;                    FLSHNUM = CLEAR;
    00FA 23          238          INX    H           ;
    00FB 3600        239          MVI    M,CLEAR     ;                    WAITLIM  = CLEAR;
    00FD 3A0000      240          LDA    LOPNUM      ;
    0100 47          241          MOV    B,A         ;
    0101 3E00        242          MVI    A,OFF       ;
    0103 CD0000      243          CALL   LOPMES      ;                    "DO LOOP MESSAGE"(OFF,LOPNUM,,,);
    0106 3A0000      244          LDA    LOPNUM      ;
    0109 47          245          MOV    B,A         ;
    010A 3EFF        246          MVI    A,ON        ;
    010C 0E00        247          MVI    C,CHKLOP    ;
    010E CD0000      248          CALL   LOPMES      ;                    "DO LOOP MESSAGE"(ON,LOPNUM,CHKLOP,,)
    0111 00          249 ENDIF44  NOP                ;                        ENDIF;
    0112 00          250 ENDIF43  NOP                ;                    ENDIF;
    0113 00          251 ENDIF42  NOP                ;                ENDIF;
    0114 2A0001      252          LHLD   STATPTR     ;
    0117 7E          253          MOV    A,M         ;
    0118 E680        254          ANI    WAITFLG     ;            IF WAITFLG = CLEAR THEN
    011A C20152      255          JNZ    ENDIF46     ;                BEGIN
    011D 23          256          INX    H           ;
    011E 34          257          INR    M           ;                WAITIM = WAITIM+1;
    011F 7E          258          MOV    A,M         ;
    0120 23          259          INX    H           ;
    0121 23          260          INX    H           ;
    0122 23          261          INX    H           ;
    0123 46          262          MOV    B,M         ;
    0124 B8          263          CMP    B           ;            IF WAITIM >= WAITLIM THEN
    0125 DA0151      264          JC     ENDIF45     ;                BEGIN
    0128 2A0001      265          LHLD   STATPTR     ;
    012B 7E          266          MOV    A,M         ;                    OFFLG   = CLEAR;
    012C E60F        267          ANI    STTMSK      ;                    ONFLG   = CLEAR;
    012E F680        268          ORI    WAITFLG     ;                    FLSHFLG = CLEAR;
    0130 77          269          MOV    M,A         ;                    WAITFLG = SET;
    0131 23          270          INX    H           ;
    0132 3600        271          MVI    M,CLEAR     ;                    OFFTIM  = CLEAR;
    0134 23          272          INX    H           ;
    0135 3600        273          MVI    M,CLEAR     ;                    FLSHTIM = CLEAR;
    0137 23          274          INX    H           ;
    0138 3600        275          MVI    M,CLEAR     ;                    FLSHNUM = CLEAR;
    013A 23          276          INX    H           ;
    013B 3600        277          MVI    M,CLEAR     ;                    WAITIM  = CLEAR;
    013D 3A0000      278          LDA    LOPNUM      ;
    0140 47          279          MOV    B,A         ;
    0141 3E00        280          MVI    A,OFF       ;
    0143 CD0000      281          CALL   LOPMES      ;                    "DO LOOP MESSAGE"(OFF,LOPNUM,,,);
    0146 3A0000      282          LDA    LOPNUM      ;
    0149 47          283          MOV    B,A         ;
    014A 3EFF        284          MVI    A,ON        ;
    014C 0E00        285          MVI    C,CHKLOP    ;
    014E CD0000      286          CALL   LOPMES      ;                    "DO LOOP MESSAGE"(ON,LOPNUM,CHKLOP,,)
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0151 00          287 ENDIF45 NOP       ;                                   ENDIF;
    0152 00          288 ENDIF46 NOP       ;                               ENDIF;
    0153 00          289 ENDIF41 NOP       ;                           ENDIF;
    0154 00          290 END4    NOP       ;                       END;
    0155 C9          291         RET       ;                   RETURN;
                     292                   ;
                     293 ********************************************************************************
                     294                   ;
                     295                   ;PROCEDURE"STATE5";
                     296                   ;   BEGIN
    0156 78          297 STATE5  MOV   A,B ;
    0157 320000      298         STA   LOPNUM;
    015A EB          299         XCHG      ;
    015B 220001      300         SHLD  STATPTR;
    015E 7E          301         MOV   A,M ;
    015F E620        302         ANI   ONFLG;
    0161 C201E9      303         JNZ   ENDIF51;
    0164 7E          304         MOV   A,M ;
    0165 E680        305         ANI   WAITFLG;
    0167 C201E9      306         JNZ   ENDIF51;        IF ONFLG = CLEAR AND WAITFLG = CLEAR THEN
    016A 7E          307         MOV   A,M ;             BEGIN
    016B E640        308         ANI   FLSHFLG;              IF FLSHFLG = CLEAR THEN
    016D C201B7      309         JNZ   ENDIF52;                BEGIN
    0170 23          310         INX   H   ;
    0171 23          311         INX   H   ;
    0172 34          312         INR   M   ;                       FLSHTIM = FLSHTIM+1;
    0173 7E          313         MOV   A,M ;
    0174 FE04        314         CPI   FLSHLIM;                    IF FLSHTIM >= FLSHLIM THEN
    0176 DA01B6      315         JC    ENDIF53;                      BEGIN
    0179 2A0001      316         LHLD  STATPTR;
    017C 7E          317         MOV   A,M ;                             OFFLG   = CLEAR;
    017D E60F        318         ANI   STTMSK;                           ONFLG   = CLEAR;
    017F F640        319         ORI   FLSHFLG;                          FLSHFLG = SET;
    0181 77          320         MOV   M,A ;                             WAITFLG = CLEAR;
    0182 23          321         INX   H   ;
    0183 23          322         INX   H   ;
    0184 23          323         INX   H   ;
    0185 34          324         INR   M   ;                       FLSHNUM = FLSHNUM+1;
    0186 7E          325         MOV   A,M ;
    0187 FE04        326         CPI   4   ;                       IF FLSHNUM >= 4 THEN
    0189 DA01B5      327         JC    ENDIF54;                      BEGIN
    018C 2A0001      328         LHLD  STATPTR;
    018F 7E          329         MOV   A,M ;                             OFFLG   = CLEAR;
    0190 E60F        330         ANI   STTMSK;                           ONFLG   = CLEAR;
    0192 F680        331         ORI   WAITFLG;                          FLSHFLG = CLEAR;
    0194 77          332         MOV   M,A ;                             WAITFLG = SET;
    0195 23          333         INX   H   ;
    0196 3600        334         MVI   M,CLEAR;                          OFFTIM  = CLEAR;
    0198 23          335         INX   H   ;
    0199 3600        336         MVI   M,CLEAR;                          FLSHTIM = CLEAR;
    019B 23          337         INX   H   ;
    019C 3600        338         MVI   M,CLEAR;                          FLSHNUM = CLEAR;
    019E 23          339         INX   H   ;
    019F 3600        340         MVI   M,CLEAR;                          WAITLIM = CLEAR;
    01A1 3A0000      341         LDA   LOPNUM;
    01A4 47          342         MOV   B,A ;
    01A5 3E00        343         MVI   A,OFF;
```

(153)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    01A7 CD0000        344          CALL    LOPMES          ;
    01AA 3A0000        345          LDA     LOPNUM          ;
    01AD 47            346          MOV     B,A             ;
    01AE 3EFF          347          MVI     A,ON            ;
    01B0 0E00          348          MVI     C,CHKLOP        ;
    01B2 CD0000        349          CALL    LOPMES          ;
    01B5 00            350 ENDIF54  NOP                     ;
    01B6 00            351 ENDIF53  NOP                     ;
    01B7 00            352 ENDIF52  NOP                     ;
    01B8 2A0001        353          LHLD    STATPTR         ;
    01BB 7E            354          MOV     A,M             ;
    01BC E680          355          ANI     WAITFLG         ;
    01BE C201E8        356          JNZ     ENDIF56         ;
    01C1 23            357          INX     H               ;
    01C2 34            358          INR     M               ;
    01C3 7E            359          MOV     A,M             ;
    01C4 FE14          360          CPI     ONLIM           ;
    01C6 DA01E7        361          JC      ENDIF55         ;
    01C9 2A0001        362          LHLD    STATPTR         ;
    01CC 7E            363          MOV     A,M             ;
    01CD E60F          364          ANI     STTMSK          ;
    01CF F620          365          ORI     ONFLG           ;
    01D1 77            366          MOV     M,A             ;
    01D2 23            367          INX     H               ;
    01D3 3600          368          MVI     M,CLEAR         ;
    01D5 23            369          INX     H               ;
    01D6 3600          370          MVI     M,CLEAR         ;
    01D8 23            371          INX     H               ;
    01D9 3600          372          MVI     M,CLEAR         ;
    01DB 23            373          INX     H               ;
    01DC 3600          374          MVI     M,CLEAR         ;
    01DE 3A0000        375          LDA     LOPNUM          ;
    01E1 47            376          MOV     B,A             ;
    01E2 3E00          377          MVI     A,OFF           ;
    01E4 CD0000        378          CALL    LOPMES          ;
    01E7 00            379 ENDIF55  NOP                     ;
    01E8 00            380 ENDIF56  NOP                     ;
    01E9 00            381 ENDIF51  NOP                     ;
    01EA 00            382 END5     NOP                     ;   END;
    01EB C9            383          RET                     ;RETURN;
                       384                                  ;
                       385                                  ;

Errors=    0
```

```
                                                                    "DO LOOP MESSAGE"(ON,LOPNUM,CHKLOP,,)
                                                                ENDIF;
                                                            ENDIF;
                                                        ENDIF;

                                                IF WAITFLG = CLEAR THEN
                                                    BEGIN

                                                        ONTIM = ONTIM+1;

                                                        IF ONTIM >= ONLIM THEN
                                                            BEGIN

                                                                OFFLG   = CLEAR;
                                                                ONFLG   = SET;
                                                                FLSHFLG = CLEAR;
                                                                WAITFLG = CLEAR;

                                                                OFFTIM  = CLEAR;

                                                                FLSHTIM = CLEAR;

                                                                FLSHNUM = CLEAR;

                                                                WAITIM  = CLEAR;


                                                                    "DO LOOP MESSAGE"(OFF,LOPNUM,,,);
                                                            ENDIF;
                                                    ENDIF;
                                            ENDIF;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1  "8085"
                      3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                      4  EXPAND
                      5  ********************************************************************
                      6  *                                                                  *
                      7  * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.           *
                      8  *                                                                  *
                      9  * PRODUCT CODE:   9-XXX-XXXXX-XXX                                 *
                     10  *                                                                  *
                     11  * MODULE:         GENMESSAGE                                      *
                     12  *                                                                  *
                     13  * FUNCTION:       THIS MODULE GENERATES THE MESSAGE FOR EACH       *
                     14  *                 TYPE OF EXTENSION, EXCHANGE AND LOOP CALL.       *
                     15  *                 THE MESSAGE IS CHECKED TO SEE IF IT HAS BEEN     *
                     16  *                 CALLED A MAX. # OF TIMES, IN WHIGH CASE THE      *
                     17  *                 SOURCE IS PLACED AT THE BACK OF THE RELEVANT     *
                     18  *                 MESSAGE STACK. TIME MESSAGES ARE GIVEN IF NO     *
                     19  *                 LAMP TYPE MESSAGES ARE THERE.                    *
                     20  *                                                                  *
                     21  * CALLS:          GENWRD,LOPMES,EXTMES,EXCMES                     *
                     22  *                                                                  *
                     23  * ON TAPE:        NOT YET                                         *
                     24  *                                                                  *
                     25  * LINKING:        LINK BLIND                                      *
                     26  * INSTRUCTIONS                                                     *
                     27  *                                                                  *
                     28  * FILE HISTORY:   STARTED 8/7/82                                  *
                     29  *                                                                  *
                     30  * 8085 CONVERSION STARTED 21/09/83                                *
                     31  *                                                                  *
                     32  * 0.0 08/07/82    INITIAL CONCEPT                                 *
                     33  *                                                                  *
                     34  *                                                                  *
                     35  ********************************************************************
                     36
                     37        GLOBAL              GENMSG,CANCEL,CHKLOP,EXCHNG,EXTENS,TIME
                     38
                     39        EXTERNAL            LOPMSTK,EXCMSTK,EXTMSTK,STTSTK,PORTB,NEWMSG
                     40        EXTERNAL            GENWRD,LOPMES,EXTMES,EXCMES,STATE2
                     41
                     42                            ;
                     43                            ;CONST:
<001F>               44 LOPMSK  EQU   00011111B    ;       LOPMSK  = 00011111B;
<001F>               45 EXCMSK  EQU   00011111B    ;       EXCMSK  = 00011111B;
<007F>               46 EXTMSK  EQU   01111111B    ;       EXTMSK  = 01111111B;
<0000>               47 OFF     EQU   00000000B    ;       OFF     = 00000000B;
<00FF>               48 ON      EQU   11111111B    ;       ON      = 11111111B;
<0040>               49 CANCEL  EQU   01000000B    ;       CANCEL  = 01000000B;
<0020>               50 CHKLOP  EQU   00100000B    ;       CHKLOP  = 00100000B;
<0010>               51 EXCHNG  EQU   00010000B    ;       EXCHNG  = 00010000B;
<0008>               52 EXTENS  EQU   00001000B    ;       EXTENS  = 00001000B;
<0004>               53 TIME    EQU   00000100B    ;       TIME    = 00000100B;
<0040>               54 NXTMSG  EQU   01000000B    ;       NXTMSG  = 01000000B;
<0000>               55 DATA    EQU   PORTB        ;       @DATA   = PORTB;
<FFFFFFFD>           56 LOPMSNM EQU   LOPMSTK-3    ;       @LOPMESNUM  = LOPMSTK-3;
<FFFFFFFD>           57 EXCMSNM EQU   EXCMSTK-3    ;       @EXCMESNUM  = EXCMSTK-3;
<FFFFFFFD>           58 EXTMSNM EQU   EXTMSTK-3    ;       @EXTMESNUM  = EXTMSTK-3;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     59                         ;
                     60           DATA          ;VAR:
0000                 61 LOPNUM    DS    1       ;       LOPNUM  : BYTE;
0001                 62 EXCNUM    DS    1       ;       EXCNUM  : BYTE;
0002                 63 EXTNUM    DS    1       ;       EXTNUM  : BYTE;
                     64                         ;
                     65           PROG          ;PROCEDURE "GENMESSAGE";
                     66                         ;   BEGIN
0000 DB00            67 GENMSG    IN    DATA    ;
0002 E640            68           ANI   NXTMSG  ;       IF NXTMSG = RESET THEN
0004 C20102          69           JNZ   ENDIF1  ;
0007 3AFFFD          70           LDA   LOPMSNM ;           BEGIN
000A B7              71           ORA   A       ;
000B CA005C          72           JZ    ENDIF2  ;
000E 3A0000          73           LDA   NEWMSG  ;               IF LOPMESNUM <> ZERO AND NEWMSG = CLEAR THEN
0011 B7              74           ORA   A       ;
0012 C2005C          75           JNZ   ENDIF2  ;
0015 3A0000          76           LDA   LOPMSTK ;                   BEGIN
0018 E61F            77           ANI   LOPMSK  ;
001A 320000          78           STA   LOPNUM  ;                       LOPNUM = ->LOPMSTK AND LOPMSK;
001D 3A0000          79           LDA   LOPMSTK ;
0020 E640            80           ANI   CANCEL  ;                       IF LOOPTYPE = CANCEL THEN
0022 CA003A          81           JZ    ELSE1   ;
0025 3A0000          82           LDA   LOPNUM  ;                           BEGIN
0028 47              83           MOV   B,A     ;
0029 3E40            84           MVI   A,CANCEL;
002B CD0000          85           CALL  GENWRD  ;                               "GO GENERATE WORDS"(CANCEL,LOPNUM,,,);
002E 3A0000          86           LDA   LOPNUM  ;
0031 47              87           MOV   B,A     ;
0032 3E00            88           MVI   A,OFF   ;
0034 CD0000          89           CALL  LOPMES  ;                               "DO LOOP MESSAGE"(OFF,LOPNUM,,,);
                     90                         ;
0037 C3005C          91 ENDIF5    JMP   ENDIF2  ;                           ENDIF;
                     92                         ;
003A 00              93 ELSE1     NOP           ;                       ELSE
003B 3A0000          94           LDA   LOPNUM  ;                           BEGIN
003E 47              95           MOV   B,A     ;
003F 3E20            96           MVI   A,CHKLOP;
0041 CD0000          97           CALL  GENWRD  ;                               "GO GENERATE WORDS"(CHKLOP,LOPNUM,,,);
0044 3A0000          98           LDA   LOPNUM  ;
0047 3D              99           DCR   A       ;
0048 47             100           MOV   B,A     ;
0049 87             101           ADD   A       ;
004A 87             102           ADD   A       ;
004B 80             103           ADD   B       ;
004C 4F             104           MOV   C,A     ;
004D 0600           105           MVI   B,00H   ;
004F 210000         106           LXI   H,STTSTK;
0052 09             107           DAD   B       ;
0053 EB             108           XCHG          ;                               STATPTR = STTSTK+(5*(LOPNUM-1));
0054 3A0000         109           LDA   LOPNUM  ;
0057 47             110           MOV   B,A     ;
0058 CD0000         111           CALL  STATE2  ;                               "DO PROCEDURE STATE2"(,LOPNUM,,STATPTR,);
005B 00             112 ENDLS1   NOP           ;                           ENDELSE;
005C 00             113 ENDIF2   NOP           ;                   ENDIF;
005D 3AFFFD         114           LDA   EXCMSNM ;
0060 B7             115           ORA   A       ;
```

(156)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0061 CA00A8       116         JZ      ENDIF3      ;
    0064 3A0000       117         LDA     NEWMSG      ;
    0067 B7           118         ORA     A           ;
    0068 C200A8       119         JNZ     ENDIF3      ;
    006B 3A0000       120         LDA     EXCMSTK     ;
    006E E61F         121         ANI     EXCMSK      ;
    0070 320001       122         STA     EXCNUM      ;
    0073 3A0000       123         LDA     EXCMSTK     ;
    0076 B7           124         ORA     A           ;
    0077 FA008B       125         JM      ELSE2       ;
                      126                             ;
    007A C680         127         ADI     10000000B   ;
    007C 320000       128         STA     EXCMSTK     ;
    007F 3A0001       129         LDA     EXCNUM      ;
    0082 47           130         MOV     B,A         ;
    0083 3E10         131         MVI     A,EXCHNG    ;
    0085 CD0000       132         CALL    GENWRD      ;
    0088 C300A8       133 ENDIF6  JMP     ENDIF3      ;
    008B 00           134 ELSE2   NOP                 ;
    008C 3A0001       135         LDA     EXCNUM      ;
    008F 47           136         MOV     B,A         ;
    0090 3E10         137         MVI     A,EXCHNG    ;
    0092 CD0000       138         CALL    GENWRD      ;
    0095 3A0001       139         LDA     EXCNUM      ;
    0098 47           140         MOV     B,A         ;
    0099 3E00         141         MVI     A,OFF       ;
    009B CD0000       142         CALL    EXCMES      ;
    009E 3A0001       143         LDA     EXCNUM      ;
    00A1 47           144         MOV     B,A         ;
    00A2 3EFF         145         MVI     A,ON        ;
    00A4 CD0000       146         CALL    EXCMES      ;
                      147                             ;
    00A7 00           148 ENDLS2  NOP                 ;
    00A8 00           149 ENDIF3  NOP                 ;
    00A9 3AFFFD       150         LDA     EXTMSNM     ;
    00AC B7           151         ORA     A           ;
    00AD CA00F4       152         JZ      ENDIF4      ;
    00B0 3A0000       153         LDA     NEWMSG      ;
    00B3 B7           154         ORA     A           ;
    00B4 C200F4       155         JNZ     ENDIF4      ;
    00B7 3A0000       156         LDA     EXTMSTK     ;
    00BA E67F         157         ANI     EXTMSK      ;
    00BC 320002       158         STA     EXTNUM      ;
    00BF 3A0000       159         LDA     EXTMSTK     ;
    00C2 B7           160         ORA     A           ;
    00C3 FA00D7       161         JM      ELSE3       ;
                      162                             ;
    00C6 C680         163         ADI     10000000B   ;
    00C8 320000       164         STA     EXTMSTK     ;
    00CB 3A0002       165         LDA     EXTNUM      ;
    00CE 47           166         MOV     B,A         ;
    00CF 3E08         167         MVI     A,EXTENS    ;
    00D1 CD0000       168         CALL    GENWRD      ;
    00D4 C300F4       169 ENDIF7  JMP     ENDIF4      ;
    00D7 00           170 ELSE3   NOP                 ;
    00D8 3A0002       171         LDA     EXTNUM      ;
    00DB 47           172         MOV     B,A         ;
```

```
IF EXCMESNUM <> ZERO AND NEWMSG = CLEAR THEN

  BEGIN

      EXCNUM = ->EXCMSTK AND EXCMSK;

      IF MSGRPTS < MAXRPTS THEN
          BEGIN

              RPTNUM = RPTNUM+1;


              "GO GENERATE WORDS"(EXCHNG,EXCNUM,,,)
          ENDIF;
      ELSE
          BEGIN

              "GO GENERATE WORDS"(EXCHNG,EXCNUM,,,);


              "DO EXCHANGE MESSAGE"(OFF,EXCNUM,,,);


              "DO EXCHANGE MESSAGE"(ON,EXCNUM,,,)

          ENDELSE;
  ENDIF;


IF EXTMESNUM <> ZERO AND NEWMSG = CLEAR THEN

  BEGIN

      EXTNUM = ->EXTMSTK AND EXTMSK;

      IF MSGRPTS < MAXRPTS THEN
          BEGIN

              RPTNUM = RPTNUM+1;


              "GO GENERATE WORDS"(EXTENS,EXCNUM,,,)
          ENDIF;
      ELSE
          BEGIN
```

```
LOCATION OBJECT CODE.LINE     SOURCE LINE

    00DC 3E08          173         MVI     A,EXTENS    ;                                "GO GENERATE WORDS"(EXTENS,EXCNUM,,,);
    00DE CD0000        174         CALL    GENWRD      ;
    00E1 3A0002        175         LDA     EXTNUM      ;
    00E4 47            176         MOV     B,A         ;
    00E5 3E00          177         MVI     A,OFF       ;
    00E7 CD0000        178         CALL    EXTMES      ;                                "DO EXTENSION MESSAGE"(OFF,EXTNUM,,,);
    00EA 3A0002        179         LDA     EXTNUM      ;
    00ED 47            180         MOV     B,A         ;
    00EE 3EFF          181         MVI     A,ON        ;
    00F0 CD0000        182         CALL    EXTMES      ;                                "DO EXTENSION MESSAGE"(ON,EXTNUM,,,)
                       183                             ;
    00F3 00            184 ENDLS3  NOP                 ;                        ENDELSE;
    00F4 00            185 ENDIF4  NOP                 ;                    ENDIF;
    00F5 3A0000        186         LDA     NEWMSG      ;
    00F8 B7            187         ORA     A           ;
    00F9 C20101        188         JNZ     ENDIF8      ;                IF NEWMSG = CLEAR THEN
                       189                             ;                    BEGIN
    00FC 3E04          190         MVI     A,TIME      ;
    00FE CD0000        191         CALL    GENWRD      ;                        "GO GENERATE WORDS"(TIME,,,,)
                       192                             ;
    0101 00            193 ENDIF8  NOP                 ;                    ENDIF;
    0102 00            194 ENDIF1  NOP                 ;            ENDIF;
    0103 00            195 END     NOP                 ;    END;
    0104 C9            196         RET                 ;RETURN;

Errors=   0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                              1 "8085"
                              3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                              4  EXPAND
                              5 ********************************************************************
                              6 *                                                                  *
                              7 * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.            *
                              8 *                                                                  *
                              9 * PRODUCT CODE:   9-XXX-XXXXX-XXX                                  *
                             10 *                                                                  *
                             11 * MODULE:         OUTVOICE                                         *
                             12 *                                                                  *
                             13 * FUNCTION:       PUTS THE BYTES IN BUFFER OUT TO DIGITALKER.      *
                             14 *                 A CHECK IS MADE TO SEE WHETHER DIGITALKER IS     *
                             15 *                 FINISHED,WHETHER NEWMESG IS SET AND IF ANY       *
                             16 *                 MORE BYTES ARE AVAILABLE.                        *
                             17 *                                                                  *
                             18 * CALLS:          NONE                                             *
                             19 *                                                                  *
                             20 * ON TAPE:        NOT YET                                          *
                             21 *                                                                  *
                             22 * LINKING:        LINK BLIND                                       *
                             23 * INSTRUCTIONS                                                     *
                             24 *                                                                  *
                             25 * FILE HISTORY:   STARTED 8/7/82                                   *
                             26 *                                                                  *
                             27 * 8085 CONVERSION STARTED 15/09/83                                 *
                             28 *                                                                  *
                             29 * 0.0 08/07/82    INITIAL CONCEPT                                  *
                             30 *                                                                  *
                             31 ********************************************************************
                             32
                             33         GLOBAL          OUTVCE,NEWMSG,BUFFER,BUFTOP,VCERDY
                             34
                             35         EXTERNAL        PORTA,PORTB,PORTC,VOICE,ENDMSG
                             36
                             37                                 ;CONST:
                             38                                 ;       SPEECH = ONE;
          <00FF>             39 SET     EQU     0FFH            ;       SET    = 0FFH;
          <00FF>             40 DISABL  EQU     11111111B       ;       DISABL = 11111111B;
          <00DF>             41 DIGEN   EQU     11011111B       ;       DIGEN  = 11011111B;
          <0080>             42 VCERDY  EQU     10000000B       ;       VCERDY = 10000000B;
          <0000>             43 DATA    EQU     PORTB           ;       @DATA  = PORT B;
                             44
                             45         DATA                    ;VAR:
0000                         46 NEWMSG  DS      1               ;       NEWMSG : BYTE;
0001                         47 WRDPTR  DS      2               ;       WRDPTR : PTR;
0003                         48 BUFFER  DS      10              ;       BUFFER : ARRAY OF SPEECH (1,,,10);
          <000D>             49 BUFTOP  EQU     $               ;
                             50                                 ;
                             51 ------------------------------- ;
                             52         PROG                    ;PROCEDURE"OUTVOICE";
                             53                                 ;   BEGIN
0000 3A0000                  54 OUTVCE  LDA     NEWMSG          ;
0003 CA0058                  55         JZ      ELSE1           ;       IF NEWMSG =SET THEN
0006 DB00                    56         IN      DATA            ;           BEGIN
0008 E680                    57         ANI     VCERDY          ;
000A CA0054                  58         JZ      ELSE2           ;               IF VOICE = READY THEN
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    000D 2A0001        59        LHLD    WRDPTR      ;                   BEGIN
    0010 7E            60        MOV     A,M         ;
    0011 FE00          61        CPI     ENDMSG      ;
    0013 CA002B        62        JZ      ELSE3       ;                       IF ->WRDPTR <> ENDMSG THEN
    0016 2A0001        63        LHLD    WRDPTR      ;                           BEGIN
    0019 3EDF          64        MVI     A,DIGEN     ;
    001B D300          65        OUT     PORTC       ;
    001D 7E            66        MOV     A,M         ;                               VOICE  = ->WRDPTR;
    001E D300          67        OUT     VOICE       ;
    0020 3EFF          68        MVI     A,DISABL    ;
    0022 D300          69        OUT     PORTC       ;
    0024 23            70        INX     H           ;                               WRDPTR = WRDPTR+1
    0025 220001        71        SHLD    WRDPTR      ;
    0028 C30051        72 ENDIF3 JMP     ENDIF2      ;                           ENDIF;
                       73                            ;                       ELSE
    002B 00            74 ELSE3  NOP                 ;                           BEGIN
    002C AF            75        XRA     A           ;
    002D 320000        76        STA     NEWMSG      ;                               NEWMSG = CLEAR;
    0030 210003        77        LXI     H,BUFFER    ;                               FOR PTR = BUFFER TO BUFTOP DO
    0033 EB            78 LOOP   XCHG                ;                                   BEGIN
    0034 21000D        79        LXI     H,BUFTOP    ;
    0037 EB            80        XCHG                ;
    0038 7C            81        MOV     A,H         ;
    0039 BA            82        CMP     D           ;
    003A C20042        83        JNZ     NOTEQ1      ;
    003D 7D            84        MOV     A,L         ;
    003E BB            85        CMP     E           ;
    003F CA0049        86        JZ      ENDFOR      ;
    0042 3E00          87 NOTEQ1 MVI     A,ENDMSG    ;
    0044 77            88        MOV     M,A         ;                                   ->PTR = ENDMSG
    0045 23            89        INX     H           ;
    0046 C30033        90        JMP     LOOP        ;
    0049 00            91 ENDFOR NOP                 ;                               ENDFOR;
    004A 210003        92        LXI     H,BUFFER    ;                               WRDPTR = BUFFER
    004D 220001        93        SHLD    WRDPTR      ;
    0050 00            94 ENDLS3 NOP                 ;                           ENDELSE;
    0051 C30055        95 ENDIF2 JMP     ENDIF1      ;                       ENDIF;
    0054 00            96 ELSE2  NOP                 ;                   ELSE
    0055 C3005F        97 ENDIF1 JMP     END         ;               ENDIF;
    0058 00            98 ELSE1  NOP                 ;           ELSE
    0059 210003        99        LXI     H,BUFFER    ;               BEGIN
    005C 220001       100        SHLD    WRDPTR      ;                   WRDPTR = BUFFER
                      101                            ;               ENDELSE;
    005F 00           102 END    NOP                 ;   END;
    0060 C9           103        RET                 ;RETURN;
                      104

Errors=    0
```

(160)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                       1  "8085"
                       3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                       4
                       5  ********************************************************************
                       6  *                                                                  *
                       7  * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.            *
                       8  *                                                                  *
                       9  * PRODUCT CODE:   9-XXX-XXXXX-XXX                                  *
                      10  *                                                                  *
                      11  * MODULE:         BINTOBCD                                         *
                      12  *                                                                  *
                      13  * FUNCTION:       THIS MODULE CONVERTS A BINARY # TO THE BCD        *
                      14  *                 EQUIVALENT IN TWO BYTES. THE BINARY # IS         *
                      15  *                 SUPPLIED IN ACCB, AND RETURNED AS TENS IN        *
                      16  *                 ACCA AND UNITS IN ACCB.                          *
                      17  *                                                                  *
                      18  * CALLS:          NONE                                             *
                      19  *                                                                  *
                      20  * ON TAPE:        NOT YET                                          *
                      21  *                                                                  *
                      22  * LINKING:        LINK BLIND                                       *
                      23  * INSTRUCTIONS                                                     *
                      24  *                                                                  *
                      25  * FILE HISTORY:   STARTED 8/7/82                                   *
                      26  *                                                                  *
                      27  * 0.0 08/07/82    INITIAL CONCEPT                                  *
                      28  *                                                                  *
                      29  * 8085 CONVERSION STARTED: 5/9/83                                  *
                      30  *                                                                  *
                      31  ********************************************************************
                      32
                      33          GLOBAL              BINBCD
                      34
                      35                              ;CONST:
                      36                              ;
                      37                              ;TYPE:
                      38                              ;
                      39          DATA                ;VAR:
                      40                              ;
                      41                              ;
                      42          PROG                ;PROCEDURE "BINTOBCD";
                      43                              ;   BEGIN
0000 50               44 BINBCD   MOV     D,B         ;       BCD  = BINNUM;
0001 58               45          MOV     E,B         ;
                      46
0002 7B               47 WHLLOOP  MOV     A,E         ;       WHILE BINNUM >= 10 DO
0003 D60A             48          SUI     0AH         ;           BEGIN
0005 5F               49          MOV     E,A         ;               BCD = BCD+6;
0006 FA0010           50          JM      ENDWHL      ;               BINNUM = BINNUM-10
0009 7A               51          MOV     A,D         ;
000A C606             52          ADI     06          ;
000C 57               53          MOV     D,A         ;
000D C30002           54          JMP     WHLLOOP     ;
0010 00               55 ENDWHL   NOP                 ;           ENDWHILE;
                      56                              ;
0011 7A               57          MOV     A,D         ;
0012 E60F             58          ANI     0FH         ;       UNITS = UNITS AND UNITSMASK;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0014 47            59         MOV     B,A         ;       UNITS = BCD;
    0015 7A            60         MOV     A,D         ;
    0016 E6F0          61         ANI     0F0H        ;       TENS = TENS AND TENSMASK THEN;
    0018 0F            62         RRC                 ;       BCD RRC 4
    0019 0F            63         RRC                 ;
    001A 0F            64         RRC                 ;
    001B 0F            65         RRC                 ;
    001C 00            66 END     NOP                 ;   END;
    001D C9            67         RET                 ;RETURN;
                       68                             ;

Errors=    0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1  "8085"
                      3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                      4  EXPAND
                      5  ********************************************************************
                      6  *                                                                  *
                      7  * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.            *
                      8  *                                                                  *
                      9  * PRODUCT CODE:   9-XXX-XXXXX-XXX                                  *
                     10  *                                                                  *
                     11  * MODULE:         TFRMSG                                           *
                     12  *                                                                  *
                     13  * FUNCTION:       THIS MODULE TRANSFERS THE MESSAGE POINTED        *
                     14  *                 TO BY "MSGPTR", INCLUDING THE DATA IN DATSTK     *
                     15  *                 AS POINTED TO BY "DATPTR", FETCHING DATA EACH    *
                     16  *                 TIME FETDATA IS SEEN, UNTIL "ENDMSG" APPEARS.    *
                     17  *                                                                  *
                     18  * CALLS:          NONE                                             *
                     19  *                                                                  *
                     20  * ON TAPE:        NOT YET                                          *
                     21  *                                                                  *
                     22  * LINKING:        LINK BLIND                                       *
                     23  * INSTRUCTIONS                                                     *
                     24  *                                                                  *
                     25  * FILE HISTORY:   STARTED 8/7/82                                   *
                     26  *                                                                  *
                     27  * 8085 CONVERSION STARTED 21/09/83                                 *
                     28  *                                                                  *
                     29  * 0.0 08/07/82    INITIAL CONCEPT                                  *
                     30  *                                                                  *
                     31  *                                                                  *
                     32  ********************************************************************
                     33
                     34          GLOBAL              TFRMSG
                     35
                     36          EXTERNAL            BUFFER,DATSTK,NEWMSG,FETDATA,ENDMSG
                     37
                     38                              ;
                     39                              ;CONST:
           <00FF>    40 SET      EQU     11111111B   ;       SET     = 11111111B;
                     41                              ;
                     42                              ;TYPE:
                     43                              ;
                     44          DATA                ;VAR:
0000                 45 BUFPTR   DS      2           ;       BUFPTR  = PTR;
0002                 46 DATPTR   DS      2           ;       DATPTR  = PTR;
0004                 47 MSGPTR   DS      2           ;       MSGPTR  = PTR;
                     48                              ;
                     49          PROG                ;PROCEDURE "GENWORDS";
                     50                              ;
0000 220004          51 TFRMSG   SHLD    MSGPTR      ;   BEGIN
0003 210000          52          LXI     H,DATSTK    ;
0006 220002          53          SHLD    DATPTR      ;       DATPTR = DATSTK;
0009 210000          54          LXI     H,BUFFER    ;
000C 220000          55          SHLD    BUFPTR      ;       BUFPTR = BUFFER;
000F 00              56 REPEAT   NOP                 ;       REPEAT
0010 2A0004          57          LHLD    MSGPTR      ;           BEGIN
0013 7E              58          MOV     A,M         ;
```

(163)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0014 FE00          59          CPI   FETDATA     ;            IF ->MSGPTR = FETCHDATA THEN
    0016 C2002B        60          JNZ   ELSE1       ;
    0019 2A0002        61          LHLD  DATPTR      ;               BEGIN
    001C 7E            62          MOV   A,M         ;
    001D 2A0000        63          LHLD  BUFPTR      ;
    0020 77            64          MOV   M,A         ;                  ->BUFPTR = ->DATPTR;
    0021 2A0002        65          LHLD  DATPTR      ;
    0024 23            66          INX   H           ;
    0025 220002        67          SHLD  DATPTR      ;                  DATPTR = DATPTR+1
    0028 C30033        68 ENDIF1   JMP   ENDLS1      ;               ENDIF;
                       69                            ;
    002B 2A0004        70 ELSE1    LHLD  MSGPTR      ;            ELSE
    002E 7E            71          MOV   A,M         ;               BEGIN
    002F 2A0000        72          LHLD  BUFPTR      ;
    0032 77            73          MOV   M,A         ;                  ->BUFPTR = ->MSGPTR
    0033 00            74 ENDLS1   NOP               ;               ENDELSE;
    0034 2A0000        75          LHLD  BUFPTR      ;
    0037 23            76          INX   H           ;
    0038 220000        77          SHLD  BUFPTR      ;            BUFPTR = BUFPTR+1;
    003B 2A0004        78          LHLD  MSGPTR      ;
    003E 23            79          INX   H           ;
    003F 220004        80          SHLD  MSGPTR      ;            MSGPTR = MSGPTR+1
    0042 00            81 ENDRPT   NOP               ;         ENDRPT;
    0043 7E            82          MOV   A,M         ;
    0044 FE00          83          CPI   ENDMSG      ;
    0046 C2000F        84          JNZ   REPEAT      ;      UNTIL ->MSGPTR = ENDMSG;
    0049 2A0000        85          LHLD  BUFPTR      ;
    004C 77            86          MOV   M,A         ;      ->BUFPTR = ENDMSG;
    004D 3EFF          87          MVI   A,SET       ;
    004F 320000        88          STA   NEWMSG      ;      NEWMSG = SET
    0052 00            89 END      NOP               ;   END;
    0053 C9            90          RET               ;RETURN;
                       91

Errors=    0
```

(164)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                              1 "8085"
                              3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                              4
                              5 ;*****************************************************************
                              6 ;                                                               *
                              7 ; PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.        *
                              8 ;                                                               *
                              9 ; PRODUCT CODE:   9-XXX-XXXXX-XXX                              *
                             10 ;                                                               *
                             11 ; MODULE:         EXTMES                                       *
                             12 ;                                                               *
                             13 ; FUNCTION:       DOES THE ACTUAL INSERTION AND REMOVAL OF THE *
                             14 ;                 EXTENSION LAMP NUMBER ON THE EXCHANGE        *
                             15 ;                 MESSAGE STACK.                               *
                             16 ;                                                               *
                             17 ; CALLS:          NONE                                         *
                             18 ;                                                               *
                             19 ; ON TAPE:        NOT YET                                      *
                             20 ;                                                               *
                             21 ; LINKING:        LINK BLIND                                   *
                             22 ; INSTRUCTIONS                                                 *
                             23 ;                                                               *
                             24 ; FILE HISTORY:   STARTED 8/7/82                               *
                             25 ;                                                               *
                             26 ; 0.0 08/07/82    INITIAL CONCEPT                              *
                             27 ;                                                               *
                             28 ; 8085 CONVERSION :STARTED 1/09/83                             *
                             29 ;                                                               *
                             30 ;*****************************************************************
                             31
                             32         GLOBAL              EXTMES,EXTMSTK
                             33
                             34 ;       EXTERNAL
                             35
                             36                             ;CONST:
<00FF>                       37 ON      EQU     0FFH        ;   ON      = 0FFH;
<00FF>                       38 SET     EQU     0FFH        ;   SET     = 0FFH;
                             39
<007F>                       40 NUMASK  EQU     01111111B   ;   NUMASK  = 01111111B;
                             41                             ;   @TODO?  = ACCA;
                             42                             ;   @EXTNUM = REGB;
                             43                             ;
                             44         DATA                ;VAR:
0000                         45 FLAG    DS      1           ;   FLAG    : BYTE;
0001                         46 TEMPTR  DS      2           ;   TEMPTR  : PTR;
0003                         47 MESNUM  DS      1           ;   MESNUM  : BYTE;
0004                         48 NXTPTR  DS      2           ;   NXTPTR  : PTR;
0006                         49 EXTMSTK DS      101         ;   EXTMSTK : ARRAY OF MESSAGE(1,,,101);
                             50                             ;
                             51                             ;TYPE:
                             52                             ;   MESSAGE : RECORD OF BITS(0,,,7)
                             53                             ;       BITS 0-6 = EXTNUM;
                             54                             ;       BIT  7   = RPTNUM;   <NUMBER OF TIMES MESSAGE HAS BEEN REPEATED>
                             55                             ;
                             56                             ;
                             57         PROG                ;PROCEDURE "EXTMES";
                             58                             ;   BEGIN
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      59 EXTMES CPI     ON          ;
0002 C2006F           60        JNZ     ELSE1       ;       IF TODO? = ON THEN
                      61                            ;          BEGIN
0005 110006           62        LXI     D,EXTMSTK   ;
0008 2A0004           63        LHLD    NXTPTR      ;              IF NXTPTR = EXTMSTK THEN
000B 7C               64        MOV     A,H         ;
000C BA               65        CMP     D           ;
000D C20022           66        JNZ     ELSE2       ;
0010 7D               67        MOV     A,L         ;
0011 BB               68        CMP     E           ;
0012 C20022           69        JNZ     ELSE2       ;
                      70                            ;                 BEGIN
0015 70               71        MOV     M,B         ;                     ->NXTPTR = EXTNUM1;
0016 23               72        INX     H           ;
0017 220004           73        SHLD    NXTPTR      ;                     NXTPTR   = NXTPTR+1;
001A 3E01             74        MVI     A,01H       ;
001C 320003           75        STA     MESNUM      ;                     MESNUM   = 1
001F C3006C           76 ENDIF2 JMP     ENDIF1      ;                 ENDIF;
0022 00               77 ELSE2  NOP                 ;              ELSE
                      78                            ;                 BEGIN
0023 AF               79        XRA     A           ;
0024 320000           80        STA     FLAG        ;                     FLAG = CLEAR;
0027 210006           81        LXI     H,EXTMSTK   ;
002A EB               82 LOOP1  XCHG                ;                     FOR PTR = EXTMSTK TO (NXTPTR-1) DO
002B 2A0004           83        LHLD    NXTPTR      ;                        BEGIN
002E EB               84        XCHG                ;
002F 7C               85        MOV     A,H         ;                           SCRATCH = ->PTR;
0030 BA               86        CMP     D           ;
0031 C20039           87        JNZ     NOTEQ1      ;
0034 7D               88        MOV     A,L         ;
0035 BB               89        CMP     E           ;
0036 CA004B           90        JZ      ENDFR1      ;
0039 7E               91 NOTEQ1 MOV     A,M         ;
003A E67F             92        ANI     NUMASK      ;                           SCRATCH = SCRATCH AND NUMASK;
003C B8               93        CMP     B           ;
003D C20046           94        JNZ     ELSE3       ;                           IF SCRATCH = EXTNUM THEN
0040 3EFF             95        MVI     A,SET       ;                              BEGIN
0042 320000           96        STA     FLAG        ;                                 FLAG = SET
0045 00               97 ENDIF3 NOP                 ;                              ENDIF;
0046 00               98 ELSE3  NOP                 ;
0047 23               99 NXTFR1 INX     H           ;
0048 C3002A          100        JMP     LOOP1       ;                        ENDFOR;
004B 00              101 ENDFR1 NOP                 ;
004C 3A0000          102        LDA     FLAG        ;
004F B7              103        ORA     A           ;
0050 C20065          104        JNZ     ELSE4       ;                     IF FLAG = CLEAR THEN
0053 2A0004          105        LHLD    NXTPTR      ;                        BEGIN
0056 70              106        MOV     M,B         ;                           ->NXTPTR = EXTNUM;
0057 23              107        INX     H           ;
0058 220004          108        SHLD    NXTPTR      ;                           NXTPTR   = NXTPTR+1;
005B 3A0003          109        LDA     MESNUM      ;                           MESNUM   = MESNUM+1
005E 3C              110        INR     A           ;
005F 320003          111        STA     MESNUM      ;
0062 C3006B          112 ENDIF4 JMP     ENDLS2      ;                        ENDIF;
0065 00              113 ELSE4  NOP                 ;                     ELSE
                     114                            ;                        BEGIN
0066 AF              115        XRA     A           ;                           FLAG = CLEAR
```

(991)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0067 320000        116          STA     FLAG        ;                                   ENDELSE;
    006A 00            117 ENDLS4   NOP                 ;                               ENDELSE;
    006B 00            118 ENDLS2   NOP                 ;                           ENDIF;
    006C C300F6        119 ENDIF1   JMP     END         ;                       ELSE
    006F 00            120 ELSE1    NOP                 ;                           BEGIN
    0070 110006        121          LXI     D,EXTMSTK   ;                               IF NXTPTR = EXTMSTK THEN
    0073 2A0004        122          LHLD    NXTPTR      ;
    0076 7C            123          MOV     A,H         ;
    0077 BA            124          CMP     D           ;
    0078 C20087        125          JNZ     ELSE5       ;
    007B 7D            126          MOV     A,L         ;
    007C BB            127          CMP     E           ;                               BEGIN
    007D C20087        128          JNZ     ELSE5       ;                                   MESNUM = ZERO
    0080 AF            129          XRA     A           ;
    0081 320003        130          STA     MESNUM      ;                               ENDIF;
    0084 C300F5        131 ENDIF5   JMP     ENDLS1      ;                           ELSE
    0087 00            132 ELSE5    NOP                 ;                               BEGIN
    0088 210006        133          LXI     H,EXTMSTK   ;
    008B EB            134 LOOP2    XCHG                ;                                   FOR PTR = EXTMSTK TO (NXTPTR-1) DO
    008C 2A0004        135          LHLD    NXTPTR      ;
    008F EB            136          XCHG                ;                                       BEGIN
    0090 7C            137          MOV     A,H         ;
    0091 BA            138          CMP     D           ;
    0092 C2009A        139          JNZ     NOTEQ2      ;
    0095 7D            140          MOV     A,L         ;
    0096 BB            141          CMP     E           ;
    0097 CA00A8        142          JZ      ENDFR2      ;                                           SCRATCH = ->PTR;
    009A 7E            143 NOTEQ2   MOV     A,M         ;                                           SCRATCH = SCRATCH AND NUMASK;
    009B E67F          144          ANI     NUMASK      ;
    009D B8            145          CMP     B           ;                                           IF SCRATCH = EXTNUM THEN
    009E C200A3        146          JNZ     ENDIF6      ;                                               BEGIN
                       147                              ;
    00A1 AF            148          XRA     A           ;                                                   ->PTR = EMPTY
    00A2 77            149          MOV     M,A         ;                                               ENDIF;
    00A3 00            150 ENDIF6   NOP                 ;
    00A4 23            151 NXTFR2   INX     H           ;
    00A5 C3008B        152          JMP     LOOP2       ;                                       ENDFOR;
    00A8 00            153 ENDFR2   NOP                 ;
    00A9 210006        154          LXI     H,EXTMSTK   ;
    00AC EB            155 LOOP3    XCHG                ;                                   FOR PTR = EXTMSTK TO (NXTPTR-1) DO
    00AD 2A0004        156          LHLD    NXTPTR      ;
    00B0 EB            157          XCHG                ;                                       BEGIN
    00B1 7C            158          MOV     A,H         ;
    00B2 BA            159          CMP     D           ;
    00B3 C200B8        160          JNZ     NOTEQ3      ;
    00B6 7D            161          MOV     A,L         ;
    00B7 BB            162          CMP     E           ;
    00B8 CA00F3        163          JZ      ENDFR3      ;
    00BB 7E            164 NOTEQ3   MOV     A,M         ;
    00BC B7            165          ORA     A           ;                                           IF ->PTR = EMPTY THEN
    00BD C200EE        166          JNZ     ELSE6       ;                                               BEGIN
    00C0 220001        167          SHLD    TEMPTR      ;
    00C3 EB            168 LOOP4    XCHG                ;                                                   FOR TEMPTR = PTR TO (NXTPTR-1) DO
    00C4 2A0004        169          LHLD    NXTPTR      ;
    00C7 EB            170          XCHG                ;                                                       BEGIN
    00C8 7C            171          MOV     A,H         ;
    00C9 BA            172          CMP     D           ;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    00CA C200D2       173         JNZ     NOTEQ4      ;
    00CD 7D           174         MOV     A,L         ;
    00CE BB           175         CMP     E           ;
    00CF CA00DA       176         JZ      ENDFR4      ;
                      177
    00D2 23           178 NOTEQ4  INX     H           ;
    00D3 7E           179         MOV     A,M         ;                          ->TEMPTR = ->TEMPTR+1
    00D4 2B           180         DCX     H           ;
    00D5 77           181         MOV     M,A         ;
    00D6 23           182 NXTFR4  INX     H           ;
    00D7 C300C3       183         JMP     LOOP4       ;
    00DA 00           184 ENDFR4  NOP                 ;                              ENDFOR;
    00DB 2A0004       185         LHLD    NXTPTR      ;
    00DE 2B           186         DCX     H           ;
    00DF 220004       187         SHLD    NXTPTR      ;                      NXTPTR = NXTPTR-1;
    00E2 3A0003       188         LDA     MESNUM      ;
    00E5 3D           189         DCR     A           ;                      MESNUM = MESNUM-1
    00E6 320003       190         STA     MESNUM      ;
    00E9 2A0001       191         LHLD    TEMPTR      ;
    00EC 2B           192         DCX     H           ;
    00ED 00           193 ENDIF7  NOP                 ;                  ENDIF;
    00EE 00           194 ELSE6   NOP                 ;
    00EF 23           195 NXTFR3  INX     H           ;
    00F0 C300AC       196         JMP     LOOP3       ;
    00F3 00           197 ENDFR3  NOP                 ;                          ENDFOR;
    00F4 00           198 ENDLS5  NOP                 ;                  ENDELSE;
    00F5 00           199 ENDLS1  NOP                 ;              ENDELSE;
    00F6 00           200 END     NOP                 ;   END;
    00F7 C9           201         RET                 ;RETURN;

Errors=    0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1 "8085"
                      3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                      4
                      5 ********************************************************************
                      6 *                                                                  *
                      7 * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.            *
                      8 *                                                                  *
                      9 * PRODUCT CODE:   9-XXX-XXXXX-XXX                                 *
                     10 *                                                                  *
                     11 * MODULE:         EXCMES                                          *
                     12 *                                                                  *
                     13 * FUNCTION:       DOES THE ACTUAL INSERTION AND REMOVAL OF THE     *
                     14 *                 EXCHANGE LAMP NUMBER ON THE EXCHANGE             *
                     15 *                 MESSAGE STACK.                                   *
                     16 *                                                                  *
                     17 * CALLS:          NONE                                            *
                     18 *                                                                  *
                     19 * ON TAPE:        NOT YET                                         *
                     20 *                                                                  *
                     21 * LINKING:        LINK BLIND                                      *
                     22 * INSTRUCTIONS                                                     *
                     23 *                                                                  *
                     24 * FILE HISTORY:   STARTED 8/7/82                                  *
                     25 *                                                                  *
                     26 * 0.0 08/07/82    INITIAL CONCEPT                                 *
                     27 *                                                                  *
                     28 * 8085 CONVERSION :STARTED 1/09/83                                *
                     29 *                                                                  *
                     30 ********************************************************************
                     31
                     32         GLOBAL               EXCMES,EXCMSTK
                     33
                     34 ;       EXTERNAL
                     35
                     36                              ;CONST:
<00FF>               37 ON      EQU      0FFH        ;   ON      = 0FFH;
<00FF>               38 SET     EQU      0FFH        ;   SET     = 0FFH;
                     39
<001F>               40 NUMASK  EQU      00011111B   ;   NUMASK  = 00011111B;
                     41                              ;   @TODO?  = ACCA;
                     42                              ;   @EXCNUM = REGB;
                     43                              ;
                     44         DATA                 ;VAR:
0000                 45 FLAG    DS       1          ;   FLAG    : BYTE;
0001                 46 TEMPTR  DS       2          ;   TEMPTR  : PTR;
0003                 47 MESNUM  DS       1          ;   MESNUM  : BYTE;
0004                 48 NXTPTR  DS       2          ;   NXTPTR  : PTR;
0006                 49 EXCMSTK DS       21         ;   EXCMSTK : ARRAY OF MESSAGE(1,,,21);
                     50                              ;
                     51                              ;TYPE:
                     52                              ;   MESSAGE : RECORD OF BITS(0,,,7)
                     53                              ;       BITS 0-4 = EXCNUM
                     54                              ;       BITS 5,6 = UNUSED
                     55                              ;       BIT  7   = RPTNUM;   <NUMBER OF TIMES MESSAGE HAS BEEN REPEATED>
                     56                              ;
                     57                              ;
                     58         PROG                 ;PROCEDURE "EXCMES";
```

(169)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      59                        ;   BEGIN
0000 FEFF             60 EXCMES CPI     ON      ;
0002 C2006F           61        JNZ     ELSE1   ;       IF TODO? = ON THEN
                      62                        ;           BEGIN
0005 110006           63        LXI     D,EXCMSTK;
0008 2A0004           64        LHLD    NXTPTR  ;               IF NXTPTR = EXCMSTK THEN
000B 7C               65        MOV     A,H     ;
000C BA               66        CMP     D       ;
000D C20022           67        JNZ     ELSE2   ;
0010 7D               68        MOV     A,L     ;
0011 BB               69        CMP     E       ;
0012 C20022           70        JNZ     ELSE2   ;
                      71                        ;                   BEGIN
0015 70               72        MOV     M,B     ;                       ->NXTPTR = EXCNUM;
0016 23               73        INX     H       ;
0017 220004           74        SHLD    NXTPTR  ;                       NXTPTR   = NXTPTR+1;
001A 3E01             75        MVI     A,01H   ;
001C 320003           76        STA     MESNUM  ;                       MESNUM   = 1
001F C3006C           77 ENDIF2 JMP     ENDIF1  ;                   ENDIF;
0022 00               78 ELSE2  NOP             ;               ELSE
                      79                        ;                   BEGIN
0023 AF               80        XRA     A       ;
0024 320000           81        STA     FLAG    ;                       FLAG = CLEAR;
0027 210006           82        LXI     H,EXCMSTK;
002A EB               83 LOOP1  XCHG            ;                       FOR PTR = EXCMSTK TO (NXTPTR-1) DO
002B 2A0004           84        LHLD    NXTPTR  ;                           BEGIN
002E EB               85        XCHG            ;
002F 7C               86        MOV     A,H     ;                               SCRATCH = ->PTR;
0030 BA               87        CMP     D       ;
0031 C20039           88        JNZ     NOTEQ1  ;
0034 7D               89        MOV     A,L     ;
0035 BB               90        CMP     E       ;
0036 CA004B           91        JZ      ENDFR1  ;
0039 7E               92 NOTEQ1 MOV     A,M     ;
003A E61F             93        ANI     NUMASK  ;                               SCRATCH = SCRATCH AND NUMASK;
003C B8               94        CMP     B       ;
003D C20046           95        JNZ     ELSE3   ;                               IF SCRATCH = EXCNUM THEN
0040 3EFF             96        MVI     A,SET   ;                                   BEGIN
0042 320000           97        STA     FLAG    ;                                       FLAG = SET
0045 00               98 ENDIF3 NOP             ;                                   ENDIF;
0046 00               99 ELSE3  NOP             ;
0047 23              100 NXTFR1 INX     H       ;
0048 C3002A          101        JMP     LOOP1   ;
004B 00              102 ENDFR1 NOP             ;                           ENDFOR;
004C 3A0000          103        LDA     FLAG    ;
004F B7              104        ORA     A       ;
0050 C20065          105        JNZ     ELSE4   ;                       IF FLAG = CLEAR THEN
0053 2A0004          106        LHLD    NXTPTR  ;                           BEGIN
0056 70              107        MOV     M,B     ;                               ->NXTPTR = EXCNUM;
0057 23              108        INX     H       ;
0058 220004          109        SHLD    NXTPTR  ;                               NXTPTR   = NXTPTR+1;
005B 3A0003          110        LDA     MESNUM  ;                               MESNUM   = MESNUM+1
005E 3C              111        INR     A       ;
005F 320003          112        STA     MESNUM  ;
0062 C3006B          113 ENDIF4 JMP     ENDLS2  ;                           ENDIF;
0065 00              114 ELSE4  NOP             ;                       ELSE
                     115                        ;                           BEGIN
```

(170)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0066 AF          116          XRA     A          ;                                   FLAG = CLEAR
    0067 320000      117          STA     FLAG       ;
    006A 00          118 ENDLS4   NOP                ;                               ENDELSE;
    006B 00          119 ENDLS2   NOP                ;                           ENDELSE;
    006C C300F6      120 ENDIF1   JMP     END        ;                       ENDIF;
    006F 00          121 ELSE1    NOP                ;                   ELSE
    0070 110006      122          LXI     D,EXCMSTK  ;                       BEGIN
    0073 2A0004      123          LHLD    NXTPTR     ;                           IF NXTPTR = EXCMSTK THEN
    0076 7C          124          MOV     A,H        ;
    0077 BA          125          CMP     D          ;
    0078 C20087      126          JNZ     ELSE5      ;
    007B 7D          127          MOV     A,L        ;
    007C BB          128          CMP     E          ;
    007D C20087      129          JNZ     ELSE5      ;                               BEGIN
    0080 AF          130          XRA     A          ;                                   MESNUM = ZERO
    0081 320003      131          STA     MESNUM     ;
    0084 C300F5      132 ENDIF5   JMP     ENDLS1     ;                               ENDIF;
    0087 00          133 ELSE5    NOP                ;                           ELSE
    0088 210006      134          LXI     H,EXCMSTK  ;                               BEGIN
    008B EB          135 LOOP2    XCHG               ;
    008C 2A0004      136          LHLD    NXTPTR     ;                                   FOR PTR = EXCMSTK TO (NXTPTR-1) DO
    008F EB          137          XCHG               ;
    0090 7C          138          MOV     A,H        ;                                       BEGIN
    0091 BA          139          CMP     D          ;
    0092 C2009A      140          JNZ     NOTEQ2     ;
    0095 7D          141          MOV     A,L        ;
    0096 BB          142          CMP     E          ;
    0097 CA00A8      143          JZ      ENDFR2     ;
    009A 7E          144 NOTEQ2   MOV     A,M        ;                                           SCRATCH = ->PTR;
    009B E61F        145          ANI     NUMASK     ;                                           SCRATCH = SCRATCH AND NUMASK;
    009D B8          146          CMP     B          ;
    009E C200A3      147          JNZ     ENDIF6     ;                                           IF SCRATCH = EXCNUM THEN
                     148                             ;                                               BEGIN
    00A1 AF          149          XRA     A          ;
    00A2 77          150          MOV     M,A        ;                                                   ->PTR = EMPTY
    00A3 00          151 ENDIF6   NOP                ;                                               ENDIF;
    00A4 23          152 NXTFR2   INX     H          ;
    00A5 C3008B      153          JMP     LOOP2      ;
    00A8 00          154 ENDFR2   NOP                ;                                       ENDFOR;
    00A9 210006      155          LXI     H,EXCMSTK  ;
    00AC EB          156 LOOP3    XCHG               ;
    00AD 2A0004      157          LHLD    NXTPTR     ;                                   FOR PTR = EXCMSTK TO (NXTPTR-1) DO
    00B0 EB          158          XCHG               ;
    00B1 7C          159          MOV     A,H        ;
    00B2 BA          160          CMP     D          ;                                       BEGIN
    00B3 C200B8      161          JNZ     NOTEQ3     ;
    00B6 7D          162          MOV     A,L        ;
    00B7 BB          163          CMP     E          ;
    00B8 CA00F3      164          JZ      ENDFR3     ;
    00BB 7E          165 NOTEQ3   MOV     A,M        ;
    00BC B7          166          ORA     A          ;
    00BD C200EE      167          JNZ     ELSE6      ;                                       IF ->PTR = EMPTY THEN
    00C0 220001      168          SHLD    TEMPTR     ;                                           BEGIN
    00C3 EB          169 LOOP4    XCHG               ;
    00C4 2A0004      170          LHLD    NXTPTR     ;                                               FOR TEMPTR = PTR TO (NXTPTR-1) DO
    00C7 EB          171          XCHG               ;
    00C8 7C          172          MOV     A,H        ;                                                   BEGIN
```

(171)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    00C9 BA          173        CMP     D          ;
    00CA C200D2      174        JNZ     NOTEQ4     ;
    00CD 7D          175        MOV     A,L        ;
    00CE BB          176        CMP     E          ;
    00CF CA00DA      177        JZ      ENDFR4     ;
                     178
    00D2 23          179 NOTEQ4 INX     H          ;                          ->TEMPTR = ->TEMPTR+1
    00D3 7E          180        MOV     A,M        ;
    00D4 2B          181        DCX     H          ;
    00D5 77          182        MOV     M,A        ;
    00D6 23          183 NXTFR4 INX     H          ;
    00D7 C300C3      184        JMP     LOOP4      ;                        ENDFOR;
    00DA 00          185 ENDFR4 NOP                ;
    00DB 2A0004      186        LHLD    NXTPTR     ;
    00DE 2B          187        DCX     H          ;
    00DF 220004      188        SHLD    NXTPTR     ;                    NXTPTR = NXTPTR-1;
    00E2 3A0003      189        LDA     MESNUM     ;
    00E5 3D          190        DCR     A          ;                    MESNUM = MESNUM-1
    00E6 320003      191        STA     MESNUM     ;
    00E9 2A0001      192        LHLD    TEMPTR     ;
    00EC 2B          193        DCX     H          ;
    00ED 00          194 ENDIF7 NOP                ;                  ENDIF;
    00EE 00          195 ELSE6  NOP                ;
    00EF 23          196 NXTFR3 INX     H          ;
    00F0 C300AC      197        JMP     LOOP3      ;
    00F3 00          198 ENDFR3 NOP                ;                ENDFOR;
    00F4 00          199 ENDLS5 NOP                ;              ENDELSE;
    00F5 00          200 ENDLS1 NOP                ;  ENDELSE;
    00F6 00          201 END    NOP                ;   END;
    00F7 C9          202        RET                ;RETURN;

Errors=    0
```

(172)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                              1 "8085"
                              3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                              4
                              5 ********************************************************************
                              6 *                                                                  *
                              7 * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.            *
                              8 *                                                                  *
                              9 * PRODUCT CODE:  9-XXX-XXXXX-XXX                                  *
                             10 *                                                                  *
                             11 * MODULE:         LOPMES                                          *
                             12 *                                                                  *
                             13 * FUNCTION:       DOES THE ACTUAL INSERTION AND REMOVAL OF THE     *
                             14 *                 LOOP NUMBER ON THE LOOP MESSAGE STACK.           *
                             15 *                                                                  *
                             16 * CALLS:          NONE                                            *
                             17 *                                                                  *
                             18 * ON TAPE:        NOT YET                                         *
                             19 *                                                                  *
                             20 * LINKING:        LINK BLIND                                      *
                             21 * INSTRUCTIONS                                                     *
                             22 *                                                                  *
                             23 * FILE HISTORY:   STARTED 8/7/82                                  *
                             24 *                                                                  *
                             25 * 0.0 08/07/82    INITIAL CONCEPT                                 *
                             26 *                                                                  *
                             27 * 8085 CONVERSION :STARTED 1/09/83                                *
                             28 *                                                                  *
                             29 ********************************************************************
                             30
                             31         GLOBAL              LOPMES,LOPMSTK
                             32
                             33 ;       EXTERNAL
                             34
                             35                             ;CONST:
<00FF>                       36 ON      EQU     0FFH        ;   ON      = 0FFH;
<00FF>                       37 SET     EQU     0FFH        ;   SET     = 0FFH;
                             38
<001F>                       39 NUMASK  EQU     00011111B   ;   NUMASK  = 00011111B;
                             40                             ;   @TODO?  = ACCA;
                             41                             ;   @LOPNUM = REGB;
                             42                             ;
                             43         DATA                ;VAR:
0000                         44 FLAG    DS      1           ;   FLAG    : BYTE;
0001                         45 TEMPTR  DS      2           ;   TEMPTR  : PTR;
0003                         46 MESNUM  DS      1           ;   MESNUM  : BYTE;
0004                         47 NXTPTR  DS      2           ;   NXTPTR  : PTR;
0006                         48 LOPMSTK DS      10          ;   LOPMSTK : ARRAY OF MESSAGE(1,,,10);
                             49                             ;
                             50                             ;TYPE:
                             51                             ;   MESSAGE : RECORD OF BITS(0,,,7)
                             52                             ;       BITS 0-4 = LOPNUM
                             53                             ;       BITS 5,6 = UNUSED
                             54                             ;       BIT  7   = RPTNUM;   <NUMBER OF TIMES MESSAGE HAS BEEN REPEATED>
                             55                             ;
                             56                             ;
                             57         PROG                ;PROCEDURE "LOPMES";
                             58                             ;   BEGIN
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

0000 FEFF          59 LOPMES CPI     ON          ;
0002 C20073        60        JNZ     ELSE1       ;       IF TODO? = ON THEN
                   61                            ;           BEGIN
0005 110006        62        LXI     D,LOPMSTK   ;
0008 2A0004        63        LHLD    NXTPTR      ;               IF NXTPTR = LOPMSTK THEN
000B 7C            64        MOV     A,H         ;
000C BA            65        CMP     D           ;
000D C20024        66        JNZ     ELSE2       ;
0010 7D            67        MOV     A,L         ;
0011 BB            68        CMP     E           ;
0012 C20024        69        JNZ     ELSE2       ;
                   70                            ;                   BEGIN
0015 79            71        MOV     A,C         ;
0016 B0            72        ORA     B           ;
0017 77            73        MOV     M,A         ;                       ->NXTPTR = LOPNUM OR TYPE;
0018 23            74        INX     H           ;
0019 220004        75        SHLD    NXTPTR      ;                       NXTPTR   = NXTPTR+1;
001C 3E01          76        MVI     A,01H       ;
001E 320003        77        STA     MESNUM      ;                       MESNUM   = 1
0021 C30070        78 ENDIF2 JMP     ENDIF1      ;                   ENDIF;
0024 00            79 ELSE2  NOP                 ;               ELSE
                   80                            ;                   BEGIN
0025 AF            81        XRA     A           ;
0026 320000        82        STA     FLAG        ;                       FLAG = CLEAR;
0029 210006        83        LXI     H,LOPMSTK   ;
002C EB            84 LOOP1  XCHG                ;                       FOR PTR = LOPMSTK TO (NXTPTR-1) DO
002D 2A0004        85        LHLD    NXTPTR      ;                           BEGIN
0030 EB            86        XCHG                ;
0031 7C            87        MOV     A,H         ;                               SCRATCH = ->PTR;
0032 BA            88        CMP     D           ;
0033 C2003B        89        JNZ     NOTEQ1      ;
0036 7D            90        MOV     A,L         ;
0037 BB            91        CMP     E           ;
0038 CA004D        92        JZ      ENDFR1      ;
003B 7E            93 NOTEQ1 MOV     A,M         ;
003C E61F          94        ANI     NUMASK      ;                               SCRATCH = SCRATCH AND NUMASK;
003E B8            95        CMP     B           ;
003F C20048        96        JNZ     ELSE3       ;                               IF SCRATCH = LOPNUM THEN
0042 3EFF          97        MVI     A,SET       ;                                   BEGIN
0044 320000        98        STA     FLAG        ;                                       FLAG = SET
0047 00            99 ENDIF3 NOP                 ;                                   ENDIF;
0048 00           100 ELSE3  NOP                 ;
0049 23           101 NXTFR1 INX     H           ;
004A C3002C       102        JMP     LOOP1       ;
004D 00           103 ENDFR1 NOP                 ;                           ENDFOR;
004E 3A0000       104        LDA     FLAG        ;
0051 B7           105        ORA     A           ;
0052 C20069       106        JNZ     ELSE4       ;                       IF FLAG = CLEAR THEN
0055 2A0004       107        LHLD    NXTPTR      ;                           BEGIN
0058 79           108        MOV     A,C         ;
0059 B0           109        ORA     B           ;
005A 77           110        MOV     M,A         ;                               ->NXTPTR = LOPNUM OR TYPE;
005B 23           111        INX     H           ;
005C 220004       112        SHLD    NXTPTR      ;                               NXTPTR   = NXTPTR+1;
005F 3A0003       113        LDA     MESNUM      ;                               MESNUM   = MESNUM+1
0062 3C           114        INR     A           ;
0063 320003       115        STA     MESNUM      ;
```

(174)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0066 C3006F        116 ENDIF4 JMP     ENDLS2       ;                       ENDIF;
    0069 00            117 ELSE4  NOP                  ;                   ELSE
                       118                             ;                       BEGIN
    006A AF            119        XRA     A            ;                           FLAG = CLEAR
    006B 320000        120        STA     FLAG         ;
    006E 00            121 ENDLS4 NOP                  ;                       ENDELSE;
    006F 00            122 ENDLS2 NOP                  ;                   ENDELSE;
    0070 C300FA        123 ENDIF1 JMP     END          ;               ENDIF;
    0073 00            124 ELSE1  NOP                  ;           ELSE
    0074 110006        125        LXI     D,LOPMSTK    ;               BEGIN
    0077 2A0004        126        LHLD    NXTPTR       ;                   IF NXTPTR = LOPMSTK THEN
    007A 7C            127        MOV     A,H          ;
    007B BA            128        CMP     D            ;
    007C C2008B        129        JNZ     ELSE5        ;
    007F 7D            130        MOV     A,L          ;
    0080 BB            131        CMP     E            ;
    0081 C2008B        132        JNZ     ELSE5        ;                       BEGIN
    0084 AF            133        XRA     A            ;                           MESNUM = ZERO
    0085 320003        134        STA     MESNUM       ;
    0088 C300F9        135 ENDIF5 JMP     ENDLS1       ;                       ENDIF;
    008B 00            136 ELSE5  NOP                  ;                   ELSE
    008C 210006        137        LXI     H,LOPMSTK    ;                       BEGIN
    008F EB            138 LOOP2  XCHG                 ;
    0090 2A0004        139        LHLD    NXTPTR       ;                           FOR PTR = LOPMSTK TO (NXTPTR-1) DO
    0093 EB            140        XCHG                 ;
    0094 7C            141        MOV     A,H          ;                               BEGIN
    0095 BA            142        CMP     D            ;
    0096 C2009E        143        JNZ     NOTEQ2       ;
    0099 7D            144        MOV     A,L          ;
    009A BB            145        CMP     E            ;
    009B CA00AC        146        JZ      ENDFR2       ;
    009E 7E            147 NOTEQ2 MOV     A,M          ;                                   SCRATCH = ->PTR;
    009F E61F          148        ANI     NUMASK       ;                                   SCRATCH = SCRATCH AND NUMASK;
    00A1 B8            149        CMP     B            ;
    00A2 C200A7        150        JNZ     ENDIF6       ;                                   IF SCRATCH = LOPNUM THEN
                       151                             ;                                       BEGIN
    00A5 AF            152        XRA     A            ;
    00A6 77            153        MOV     M,A          ;                                           ->PTR = EMPTY
    00A7 00            154 ENDIF6 NOP                  ;                                       ENDIF;
    00A8 23            155 NXTFR2 INX     H            ;
    00A9 C3008F        156        JMP     LOOP2        ;
    00AC 00            157 ENDFR2 NOP                  ;                               ENDFOR;
    00AD 210006        158        LXI     H,LOPMSTK    ;
    00B0 EB            159 LOOP3  XCHG                 ;
    00B1 2A0004        160        LHLD    NXTPTR       ;                           FOR PTR = LOPMSTK TO (NXTPTR-1) DO
    00B4 EB            161        XCHG                 ;
    00B5 7C            162        MOV     A,H          ;                               BEGIN
    00B6 BA            163        CMP     D            ;
    00B7 C200BF        164        JNZ     NOTEQ3       ;
    00BA 7D            165        MOV     A,L          ;
    00BB BB            166        CMP     E            ;
    00BC CA00F7        167        JZ      ENDFR3       ;
    00BF 7E            168 NOTEQ3 MOV     A,M          ;
    00C0 B7            169        ORA     A            ;
    00C1 C200F2        170        JNZ     ELSE6        ;                                   IF ->PTR = EMPTY THEN
    00C4 220001        171        SHLD    TEMPTR       ;                                       BEGIN
    00C7 EB            172 LOOP4  XCHG                 ;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    00C8 2A0004       173         LHLD    NXTPTR      ;                FOR TEMPTR = PTR TO (NXTPTR-1) DO
    00CB EB           174         XCHG                ;
    00CC 7C           175         MOV     A,H         ;                   BEGIN
    00CD BA           176         CMP     D           ;
    00CE C200D6       177         JNZ     NOTEQ4      ;
    00D1 7D           178         MOV     A,L         ;
    00D2 BB           179         CMP     E           ;
    00D3 CA00DE       180         JZ      ENDFR4      ;
                      181
    00D6 23           182 NOTEQ4  INX     H           ;
    00D7 7E           183         MOV     A,M         ;                      ->TEMPTR = ->TEMPTR+1
    00D8 2B           184         DCX     H           ;
    00D9 77           185         MOV     M,A         ;
    00DA 23           186 NXTFR4  INX     H           ;
    00DB C300C7       187         JMP     LOOP4       ;
    00DE 00           188 ENDFR4  NOP                 ;                   ENDFOR;
    00DF 2A0004       189         LHLD    NXTPTR      ;
    00E2 2B           190         DCX     H           ;
    00E3 220004       191         SHLD    NXTPTR      ;                NXTPTR = NXTPTR-1;
    00E6 3A0003       192         LDA     MESNUM      ;
    00E9 3D           193         DCR     A           ;                MESNUM = MESNUM-1
    00EA 320003       194         STA     MESNUM      ;
    00ED 2A0001       195         LHLD    TEMPTR      ;
    00F0 2B           196         DCX     H           ;
    00F1 00           197 ENDIF7  NOP                 ;             ENDIF;
    00F2 00           198 ELSE6   NOP                 ;
    00F3 23           199 NXTFR3  INX     H           ;
    00F4 C300B0       200         JMP     LOOP3       ;                         ENDFOR;
    00F7 00           201 ENDFR3  NOP                 ;                   ENDELSE;
    00F8 00           202 ENDLS5  NOP                 ;
    00F9 00           203 ENDLS1  NOP                 ;          ENDELSE;
    00FA 00           204 END     NOP                 ;    END;
    00FB C9           205         RET                 ;RETURN;

Errors=    0
```

(176)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0009 320006        59          STA    DATSTK      ;        BEGIN
    000C 320007        60          STA    DATSTK+1    ;
    000F 320008        61          STA    DATSTK+2    ;            ->PTR = ENDMSG
    0012 320009        62          STA    DATSTK+3    ;
    0015 00            63 ENDFOR   NOP                ;        ENDFOR;
    0016 210006        64          LXI    H,DATSTK    ;
    0019 220004        65          SHLD   DATPTR      ;    DATPTR = DATSTK;
    001C 3A0000        66          LDA    TYPE        ;
    001F FE00          67          CPI    TIME        ;    IF TYPE = TIME THEN
    0021 C2003F        68          JNZ    ELSE1       ;
    0024 3A0000        69          LDA    HOURS       ;        BEGIN
    0027 2A0004        70          LHLD   DATPTR      ;
    002A 77            71          MOV    M,A         ;            ->DATPTR = HOURS;
    002B 23            72          INX    H           ;
    002C 220004        73          SHLD   DATPTR      ;            DATPTR = DATPTR+1;
    002F 3A0000        74          LDA    MINUTES     ;
    0032 47            75          MOV    B,A         ;
    0033 3A0000        76          LDA    TYPE        ;
    0036 2A0004        77          LHLD   DATPTR      ;
    0039 CD0000        78          CALL   BINDAT      ;            "GO CONVERT BIN TO DATA"(TYPE,MINUTES,,,DATPTR)
    003C C3004C        79 ENDIF1   JMP    ENDLS1      ;        ENDIF;
                       80                             ;
    003F 3A0001        81 ELSE1    LDA    BINNUM      ;    ELSE
    0042 47            82          MOV    B,A         ;
    0043 3A0000        83          LDA    TYPE        ;        BEGIN
    0046 2A0004        84          LHLD   DATPTR      ;
    0049 CD0000        85          CALL   BINDAT      ;            "GO CONVERT BIN TO DATA"(TYPE,BINNUM,,,DATPTR)
    004C 00            86 ENDLS1   NOP                ;        ENDELSE;
                       87                             ;
    004D 3A0000        88 CASE     LDA    TYPE        ;    CASE TYPE OF
                       89                             ;
    0050 FE00          90 CASEXT   CPI    EXTENS      ;        EXTENS:
    0052 C2005E        91          JNZ    CASEXC      ;            BEGIN
    0055 210000        92          LXI    H,EXTMSG    ;
    0058 220002        93          SHLD   MSGPTR      ;                MSGPTR = EXTMSG
    005B C30090        94 ENDEXT   JMP    ENDCASE     ;            ENDEXTENS;
                       95                             ;
    005E FE00          96 CASEXC   CPI    EXCHNG      ;        EXCHNG:
    0060 C2006C        97          JNZ    CASCAN      ;            BEGIN
    0063 210000        98          LXI    H,EXCMSG    ;
    0066 220002        99          SHLD   MSGPTR      ;                MSGPTR = EXCMSG
    0069 C30090       100          JMP    ENDCASE     ;            ENDEXCHNG;
                      101                             ;
    006C FE00         102 CASCAN   CPI    CANCEL      ;        CANCEL:
    006E C2007A       103          JNZ    CASCHK      ;            BEGIN
    0071 210000       104          LXI    H,CANMSG    ;
    0074 220002       105          SHLD   MSGPTR      ;                MSGPTR = CANMSG
    0077 C30090       106          JMP    ENDCASE     ;            ENDCANCEL;
                      107                             ;
    007A FE00         108 CASCHK   CPI    CHKLOP      ;        CHKLOP:
    007C C20088       109          JNZ    CASTIM      ;            BEGIN
    007F 210000       110          LXI    H,CHKMSG    ;
    0082 220002       111          SHLD   MSGPTR      ;                MSGPTR = CHKMSG
    0085 C30090       112          JMP    ENDCASE     ;            ENDCHKLOP;
                      113                             ;
    0088 00           114 CASTIM   NOP                ;        TIME:
    0089 210000       115          LXI    H,TIMMSG    ;            BEGIN
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    008C 220002       116          SHLD   MSGPTR       ;                  MSGPTR = TIMMSG
    008F 00           117 ENDTIM   NOP                  ;               ENDTIME;
    0090 00           118 ENDCASE  NOP                  ;       ENDCASE;
    0091 2A0002       119          LHLD   MSGPTR        ;
    0094 CD0000       120          CALL   TERMSG        ;       "GO TRANSFER MESSAGE"(,,,,MSGPTR)
    0097 00           121 END      NOP                  ;   END;
    0098 C9           122          RET                  ;RETURN;
                      123

Errors=    0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1  "8085"
                      3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                      4  EXPAND
                      5  ********************************************************************
                      6  *                                                                  *
                      7  * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.           *
                      8  *                                                                  *
                      9  * PRODUCT CODE:   9-XXX-XXXXX-XXX                                 *
                     10  *                                                                  *
                     11  * MODULE:         BINTODAT                                        *
                     12  *                                                                  *
                     13  * FUNCTION:       THIS MODULE CONVERTS THE BINARY`$ OF THE        *
                     14  *                 EXTENSION,EXCHANGE,LOOP OR MINUTES TO THE       *
                     15  *                 APPROPRIATE DATA AS REQUIRED BY THE MESSAGE     *
                     16  *                 TYPE E.G. "MINUTES OF TIME" OR "OTHER".         *
                     17  *                                                                  *
                     18  * CALLS:          BINTOBCD                                        *
                     19  *                                                                  *
                     20  * ON TAPE:        NOT YET                                         *
                     21  *                                                                  *
                     22  * LINKING:        LINK BLIND                                      *
                     23  * INSTRUCTIONS                                                     *
                     24  *                                                                  *
                     25  * FILE HISTORY:   STARTED 8/7/82                                  *
                     26  *                                                                  *
                     27  * 8085 CONVERSION STARTED 22/09/83                                *
                     28  *                                                                  *
                     29  * 0.0 08/07/82    INITIAL CONCEPT                                 *
                     30  *                                                                  *
                     31  *                                                                  *
                     32  ********************************************************************
                     33
                     34          GLOBAL              BINDAT
                     35
                     36          EXTERNAL            SPACE20,HUNDRED,TIME,BASE
                     37          EXTERNAL            BINBCD,0
                     38
                     39                              ;CONST:
                     40                              ;
                     41                              ;TYPE:
                     42                              ;
                     43                              ;
                     44          DATA                ;VAR:
0000                 45 TYPE     DS      1           ;       TYPE    = BYTE;
0001                 46 BINNUM   DS      1           ;       BINNUM  = BYTE;
0002                 47 DATPTR   DS      2           ;       DATPTR  = PTR;
                     48                              ;
                     49                              ;
                     50          PROG                ;PROCEDURE "BINTODAT";
                     51                              ;
0000 320000          52 BINDAT   STA     TYPE        ;   BEGIN
0003 78              53          MOV     A,B         ;
0004 320001          54          STA     BINNUM      ;
0007 220002          55          SHLD    DATPTR      ;
000A 3A0000          56          LDA     TYPE        ;
000D FE00            57          CPI     TIME        ;       IF TYPE = TIME THEN
000F C20089          58          JNZ     ELSE1       ;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

0012 3A0001          59         LDA   BINNUM      ;          BEGIN
0015 FE00            60         CPI   00          ;
0017 C2002A          61         JNZ   ELSE2       ;              IF BINNUM = 00 THEN
001A 3E00            62         MVI   A,HUNDRED   ;                BEGIN
001C 2A0002          63         LHLD  DATPTR      ;
001F 77              64         MOV   M,A         ;                    ->DATPTR = 'HUNDRED';
0020 23              65         INX   H           ;
0021 220002          66         SHLD  DATPTR      ;                    DATPTR   = DATPTR+1;
0024 3E00            67         MVI   A,SPACE20   ;
0026 77              68         MOV   M,A         ;                    ->DATPTR = SPACE20
0027 C30085          69 ENDIF2  JMP   ENDLS2      ;                ENDIF;
002A 3A0001          70 ELSE2   LDA   BINNUM      ;              ELSE
002D FE0A            71         CPI   10          ;                BEGIN
002F D20043          72         JNC   ELSE3       ;                    IF BINNUM <= 9 THEN
0032 3E00            73         MVI   A,0         ;                      BEGIN
0034 2A0002          74         LHLD  DATPTR      ;
0037 77              75         MOV   M,A         ;                          ->DATPTR = '0';
0038 23              76         INX   H           ;
0039 220002          77         SHLD  DATPTR      ;                          DATPTR   = DATPTR+1;
003C 3A0001          78         LDA   BINNUM      ;
003F 77              79         MOV   M,A         ;                          ->DATPTR = BINNUM
0040 C30084          80 ENDIF3  JMP   ENDLS3      ;                      ENDIF;
0043 3A0001          81 ELSE3   LDA   BINNUM      ;                    ELSE
0046 FE15            82         CPI   21          ;                      BEGIN
0048 D2005C          83         JNC   ELSE4       ;                          IF BINNUM <= 20 THEN
004B 3A0001          84         LDA   BINNUM      ;                            BEGIN
004E 2A0002          85         LHLD  DATPTR      ;
0051 77              86         MOV   M,A         ;                              ->DATPTR = BINNUM;
0052 23              87         INX   H           ;
0053 220002          88         SHLD  DATPTR      ;                              DATPTR   = DATPTR+1;
0056 3E00            89         MVI   A,SPACE20   ;
0058 77              90         MOV   M,A         ;                              ->DATPTR = SPACE20
0059 C30083          91 ENDIF4  JMP   ENDLS4      ;                            ENDIF;
005C 00              92 ELSE4   NOP               ;                          ELSE
005D 3A0001          93         LDA   BINNUM      ;                            BEGIN
0060 47              94         MOV   B,A         ;
0061 CD0000          95         CALL  BINBCD      ;                              "GO CONVERT BIN TO BCD"
0064 C600            96         ADI   BASE        ;                                (,BINNUM,,,)(TENS,UNITS,,,);
0066 2A0002          97         LHLD  DATPTR      ;
0069 77              98         MOV   M,A         ;                              ->DATPTR = TENS+BASE;
006A 23              99         INX   H           ;
006B 220002         100         SHLD  DATPTR      ;                              DATPTR   = DATPTR+1;
006E 78             101         MOV   A,B         ;
006F FE00           102         CPI   00          ;
0071 C2007D         103         JNZ   ELSE5       ;                              IF UNITS = 00 THEN
0074 3E00           104         MVI   A,SPACE20   ;
0076 2A0002         105         LHLD  DATPTR      ;                                BEGIN
0079 77             106         MOV   M,A         ;                                  ->DATPTR = SPACE20
007A C30082         107 ENDIF5  JMP   ENDLS5      ;                                ENDIF;
007D 78             108 ELSE5   MOV   A,B         ;                              ELSE
007E 2A0002         109         LHLD  DATPTR      ;                                BEGIN
0081 77             110         MOV   M,A         ;                                  ->DATPTR = UNITS
0082 00             111 ENDLS5  NOP               ;                                ENDELSE;
0083 00             112 ENDLS4  NOP               ;                            ENDELSE;
0084 00             113 ENDLS3  NOP               ;                          ENDELSE;
0085 00             114 ENDLS2  NOP               ;                      ENDELSE;
0086 C300EA         115 ENDIF1  JMP   END         ;          ENDIF;
```

(181)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0089 3A0001       116 ELSE1   LDA     BINNUM      ;       ELSE
    008C FE64         117         CPI     100         ;           BEGIN
    008E C200A1       118         JNZ     ELSE6       ;               IF BINNUM = 100 THEN
    0091 3E00         119         MVI     A,HUNDRED   ;                   BEGIN
    0093 2A0002       120         LHLD    DATPTR      ;
    0096 77           121         MOV     M,A         ;                       ->DATPTR = 'HUNDRED';
    0097 23           122         INX     H           ;
    0098 220002       123         SHLD    DATPTR      ;                       DATPTR   = DATPTR+1;
    009B 3E00         124         MVI     A,SPACE20   ;
    009D 77           125         MOV     M,A         ;                       ->DATPTR = SPACE20;
    009E C300E8       126 ENDIF6  JMP     ENDLS6      ;                   ENDIF;
    00A1 3A0001       127 ELSE6   LDA     BINNUM      ;               ELSE
    00A4 FE15         128         CPI     21          ;                   BEGIN
    00A6 D200BA       129         JNC     ELSE7       ;                       IF BINNUM <= 20 THEN
    00A9 3A0001       130         LDA     BINNUM      ;                           BEGIN
    00AC 2A0002       131         LHLD    DATPTR      ;
    00AF 77           132         MOV     M,A         ;                               ->DATPTR = BINNUM;
    00B0 23           133         INX     H           ;
    00B1 220002       134         SHLD    DATPTR      ;                               DATPTR   = DATPTR+1;
    00B4 3E00         135         MVI     A,SPACE20   ;
    00B6 77           136         MOV     M,A         ;                               ->DATPTR = SPACE20
    00B7 C300E7       137 ENDIF7  JMP     ENDLS7      ;                           END IF;
    00BA 00           138 ELSE7   NOP                 ;                       ELSE
    00BB 3A0001       139         LDA     BINNUM      ;                           BEGIN
    00BE 47           140         MOV     B,A         ;                               "GO CONVERT BIN TO BCD"
    00BF CD0000       141         CALL    BINBCD      ;                                   (,BINNUM,,,)(TENS,UNITS,,,);
    00C2 4F           142         MOV     C,A         ;
    00C3 78           143         MOV     A,B         ;
    00C4 FE00         144         CPI     00          ;
    00C6 C200DA       145         JNZ     ELSE8       ;                               IF UNITS = 00 THEN
    00C9 79           146         MOV     A,C         ;
    00CA C600         147         ADI     BASE        ;                                   BEGIN
    00CC 2A0002       148         LHLD    DATPTR      ;
    00CF 77           149         MOV     M,A         ;                                       ->DATPTR = TENS+BASE;
    00D0 23           150         INX     H           ;
    00D1 220002       151         SHLD    DATPTR      ;                                       DATPTR   = DATPTR+1;
    00D4 3E00         152         MVI     A,SPACE20   ;
    00D6 77           153         MOV     M,A         ;                                       ->DATPTR = SPACE20
    00D7 C300E6       154 ENDIF8  JMP     ENDLS8      ;                                   ENDIF;
    00DA 00           155 ELSE8   NOP                 ;                               ELSE
    00DB 79           156         MOV     A,C         ;
    00DC 2A0002       157         LHLD    DATPTR      ;                                   BEGIN
    00DF 77           158         MOV     M,A         ;                                       ->DATPTR = TENS;
    00E0 23           159         INX     H           ;
    00E1 220002       160         SHLD    DATPTR      ;                                       DATPTR   = DATPTR+1;
    00E4 78           161         MOV     A,B         ;
    00E5 77           162         MOV     M,A         ;                                       ->DATPTR = UNITS
    00E6 00           163 ENDLS8  NOP                 ;                                   ENDELSE;
    00E7 00           164 ENDLS7  NOP                 ;                           ENDELSE;
    00E8 00           165 ENDLS6  NOP                 ;                   ENDELSE;
    00E9 00           166 ENDLS1  NOP                 ;           ENDELSE;
    00EA 00           167 END     NOP                 ;   END;
    00EB C9           168         RET                 ;RETURN;

Errors=    0
```

(182)

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1  "8085"
                      3  NAME  "9-XXX-XXXXX-XXX ISS 1"
                      4  EXPAND
                      5  ********************************************************************
                      6  *                                                                  *
                      7  * PRODUCT:        BLIND OPERATOR CONSOLE AID FOR U.F.P.           *
                      8  *                                                                  *
                      9  * PRODUCT CODE:   9-XXX-XXXXX-XXX                                 *
                     10  *                                                                  *
                     11  * MODULE:         MESSAGES                                        *
                     12  *                                                                  *
                     13  * FUNCTION:       THIS MODULE DEFINES THE MESSAGES FOR THE        *
                     14  *                 VARIOUS CONDITIONS THAT "OUTVOICE" SIGNALS      *
                     15  *                 TO THE BLIND OPERATOR.                          *
                     16  *                                                                  *
                     17  * CALLS:          NONE                                            *
                     18  *                                                                  *
                     19  * ON TAPE:        NOT YET                                         *
                     20  *                                                                  *
                     21  * LINKING:        LINK BLIND                                      *
                     22  * INSTRUCTIONS                                                     *
                     23  *                                                                  *
                     24  * FILE HISTORY:   STARTED 8/7/82                                  *
                     25  *                                                                  *
                     26  * 8085 CONVERSION STARTED 20/09/83                                *
                     27  *                                                                  *
                     28  * 0.0 08/07/82    INITIAL CONCEPT                                 *
                     29  *                                                                  *
                     30  *                                                                  *
                     31  ********************************************************************
                     32
                     33           GLOBAL          EXTMSG,EXCMSG,CHKMSG,CANMSG,TIMMSG
                     34           GLOBAL          HRSMSG,MINMSG,BCDMSG
                     35           GLOBAL          HITONE,LOWTONE,FETDATA,ENDMSG
                     36           GLOBAL          SPACE20,ZERO,HUNDRED,BASE,0
                     37
<00FD>               38 FETDATA   EQU   0FDH
<00FF>               39 ENDMSG    EQU   0FFH
<0037>               40 EX        EQU   037H
<002B>               41 EL        EQU   02BH
<00BF>               42 WAIT      EQU   0BFH
<005E>               43 IN        EQU   05EH
<003E>               44 CANCEL    EQU   03EH
<008B>               45 TIME      EQU   08BH
<0060>               46 IS        EQU   060H
<0043>               47 SPACE20   EQU   043H
<0044>               48 SPACE40   EQU   044H
<0045>               49 SPACE80   EQU   045H
<0046>               50 SPACE160  EQU   046H
<0047>               51 SPACE320  EQU   047H
<001F>               52 ZERO      EQU   01FH
<001C>               53 HUNDRED   EQU   01CH
<0041>               54 HITONE    EQU   041H
<0042>               55 LOWTONE   EQU   042H
<0012>               56 BASE      EQU   012H
<006E>               57 MINUTE    EQU   06EH
<005D>               58 HOUR      EQU   05DH
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

             <0083>     59 SET      EQU   083H
             <0081>     60 SS       EQU   081H
             <002E>     61 O        EQU   02EH
                        62
    0000 3744FD43FD     63 EXTMSG   DB    EX,SPACE40,FETDATA,SPACE20,FETDATA,ENDMSG
    0005 FF
    0006 5E44FD43FD     64 EXCMSG   DB    IN,SPACE40,FETDATA,SPACE20,FETDATA,ENDMSG
    0008 FF
    000C 2B44FD43FD     65 CHKMSG   DB    EL,SPACE40,FETDATA,SPACE20,FETDATA,WAIT,ENDMSG
    0011 8FFF
    0013 3E442B43FD     66 CANMSG   DB    CANCEL,SPACE40,EL,SPACE20,FETDATA,ENDMSG
    0018 FF
    0019 8B6044FD43     67 TIMMSG   DB    TIME,IS,SPACE40,FETDATA,SPACE20,FETDATA,SPACE20,FETDATA,ENDMSG
    001E FD43FDFF
    0022 83445D81FF     68 HRSMSG   DB    SET,SPACE40,HOUR,SS,ENDMSG
    0027 83446E81FF     69 MINMSG   DB    SET,SPACE40,MINUTE,SS,ENDMSG
    002C FDFDFF         70 BCDMSG   DB    FETDATA,FETDATA,ENDMSG
                        71
                        72
                        73

Errors=   0
```