

ROYAL OBSERVATORY, HONG KONG

Technical Note No. 84

REAL-TIME EXCHANGE OF DIGITAL RADAR IMAGES
BETWEEN HONG KONG AND GUANGZHOU

by

YEUNG Kai-hing

CHAN Ka-wah

CROWN COPYRIGHT RESERVED

Published April 1991

Prepared by :

Royal Observatory
134A Nathan Road
Kowloon
Hong Kong

This report is prepared and disseminated in the interest of promoting information exchange. The findings, conclusion and views contained herein are those of the authors and not necessarily those of the Royal Observatory or the Government of Hong Kong.

The Government of Hong Kong (including its officers and employees) makes no warranty or representation, expressed or implied, or assumes any legal liability or responsibility (including liability for negligence) for the accuracy, completeness, or usefulness of the information contained herein or for the loss, damage, or injury (including death) which may result, whether directly or indirectly, from the supply or use of such information.

Mention of product or manufacturer does not necessarily constitute or imply endorsement or recommendation.

Permission to reproduce any part of this publication should be obtained through the Royal Observatory.

621.397

SUMMARY

This manual contains functional descriptions of the hardware and software of a radar image exchange system developed by the Royal Observatory for the real-time exchange of radar images between the Royal Observatory, Hong Kong and the Guangdong Meteorological Bureau, Guangzhou, PRC. Procedures for operating the system are also described in this manual.

CONTENTS

SUMMARY	ii
CONTENTS	iii
FIGURES	iv
1. INTRODUCTION	1
2. SYSTEM CONFIGURATION	2
3. COMMUNICATION PROTOCOL AND COMMANDS	4
3.1. Communication protocols	4
3.2. Time-outs and re-tries	7
3.3. Commands	7
4. SOFTWARE FUNCTIONAL DESCRIPTIONS	9
4.1. Overview	9
4.2. Automatic transmission of radar images	11
4.3. Reception of files	12
4.4. Image display	12
4.5. Directory listing	12
4.6. Manual file transfer	13
4.7. Dialogue	13
5. PROGRAM MODULE DESCRIPTION	15
5.1. Overview	14
5.2. Program module description	14
5.3. Program call tree	23
5.4. Program compilation and linking	23
5.5. Program limitation	23
6. OPERATION PROCEDURES	25
6.1. Overview	25
6.2. Operator interface menu	25
6.3. Examining history of telecommunication activities	28
6.4. Data archival	28
7. CONCLUDING REMARKS	29
REFERENCE	30
APPENDIX A - Radar Image Data Format	43
A.1. Hong Kong radar image format	43
A.2. Guangzhou radar image format	44
APPENDIX B - Program call tree	47

FIGURES

Figure 1. System block diagram	31
Figure 2. Cable connections to the microcomputers	32
Figure 3. Transfer set up sequence	33
Figure 4. Data transfer sequence	34
Figure 5. Data break recovery sequence	35
Figure 6. Termination sequence	36
Figure 7. No DATA frame time-out recovery sequence	37
Figure 8. No WINDOW frame time-out recovery sequence	38
Figure 9. Operator interface menu	39
Figure 10. Data flow diagram for the microcomputer at Hong Kong	40
Figure 11. Data flow diagram for the microcomputer at Guangzhou	41
Figure 12. Operator interface menu	42

1. INTRODUCTION

The meteorological telecommunication circuit between Hong Kong and Beijing was first set up in December 1975, as a regional circuit of the Global Telecommunication System under the World Weather Watch in Regional Association II. It had a speed of 75 bits per second (bps) only and was used for exchange of conventional meteorological data in alphanumeric codes.

By the late 1980s, the volume of meteorological data exchanged over the Hong Kong - Beijing circuit had increased many folds and reached the ultimate capacity of this low speed circuit. In February 1990, the Director of the Royal Observatory, Hong Kong and the Director of the National Meteorological Centre signed an memorandum of understanding (reference 1) to upgrade the Hong Kong - Beijing meteorological circuit to 9 600 bps with effect from 15th of that month. The circuit is made up of two segments, i.e. Hong Kong - Guangzhou, and Guangzhou - Beijing. The upgrading was not only intended to cope with the increasing data volume, but to allow exchange of new classes of meteorological data as well.

In the upgraded circuit, the 9 600 bps modems at Hong Kong and Guangzhou are each configured to operate with three full duplex synchronous ports which are used as follows :

- Port A - facsimile transmission at 4 800 bps
- Port B - data transmission at 2 400 bps under LAPB of CCITT X.25 protocol
- Port C - real-time exchange of digital radar images between Hong Kong and Guangzhou at 2 400 bps after conversion from asynchronous to synchronous transmission by an async/sync converter.

The exchange of radar images between Hong Kong and Guangzhou is carried out through two personal microcomputer connected to the radar processors at the two sides. The communication protocol for image exchange was jointly designed by the Royal Observatory and the Guangdong Meteorological Bureau. The software for exchanging, displaying and archiving radar images and related data was developed by the Royal Observatory with input from the Guangdong Meteorological Bureau. Specifically, the latter provided the software modules for decoding and displaying radar images of the Guangzhou radar. The Academy of Meteorological Sciences of the State Meteorological Administration, PRC also contributed to the project by developing the software for transferring radar images from the radar system at Guangzhou to the microcomputer there. The hardware and software used for real-time exchange of radar images between Hong Kong and Guangzhou are described in this manual. The procedures for operation are also described.

2. SYSTEM CONFIGURATION

The radar image exchange system comprises the following hardware :

At the Royal Observatory, Hong Kong

- (a) Vitro MR-790S 10 cm radar
- (b) Data General Eclipse S/140 radar processor
- (c) AST Premium/286 microcomputer
- (d) Racal-Milgo Omnimode 96 modem with one 4 800 bps and two 2 400 bps full duplex synchronous channels
- (e) General DataComm ASC-1 async/sync converter

At the Guangdong Meteorological Bureau, Guangzhou

- (f) 714B radar
- (g) IS-68K radar processor
- (h) AST Premium/286 microcomputer
- (i) Racal-Milgo Omnimode 96 modem with one 4 800 bps and two 2 400 bps full duplex synchronous channels
- (j) General DataComm ASC-1 Async/Sync converter

The above hardware are configured as shown in Figure 1. The exchange of radar images is controlled through the two microcomputers, which also serve as display stations.

The radar processors at Hong Kong and Guangzhou can both generate a number of display products including the 3 km Constant Altitude Plane Position Indicator (CAPPI). The data formats of the different types of radar image files are given in Appendix A.

The microcomputers at Hong Kong and Guangzhou are each an IBM PC/AT compatible AST Premium/286 system unit operating at 10 MHz, which includes :

- (a) One 40 MB hard disk drive, partitioned into logical drives C and D
- (b) One serial port, disabled
- (c) One printer port
- (d) 2 MB main memory
- (e) One 1.2 MB floppy disk drive
- (f) One 360 kB floppy disk drive
- (g) One VGA adapter and VGA monitor

- (h) One serial interface card (Model CT-450 multi I/O card at Hong Kong; 4-serial-port extension board at Guangzhou). Addresses and interrupts of COM1 and COM2 were set as indicated in Figure 1.

In addition, the microcomputer at Guangzhou has a 3M BC503 ethernet card which is connected to other data processing and display systems in the Guangdong Meteorological Bureau.

The cable connections to the microcomputers are shown in Figure 2.

The disk operating system on the microcomputer is Microsoft MS-DOS Version 3.3. The CONFIG.SYS file has the line FILES=20 (or greater number).

The operational radar image exchange software on the microcomputer at Hong Kong consists of three programs, namely MODE.COM (DOS command to set serial port line speed and byte format), LCOM.EXE (serial port driver), and WSHK.EXE (main program) in a directory called RADARXCH; while that on the microcomputer at Guangzhou consists of MODE.COM, LCOM.EXE, and WSGZ.EXE also in a directory called RADARXCH. WSHK.EXE and WSGZ.EXE are essentially the same apart from station name definitions.

In routine operation, the Eclipse S/140 radar processor at Hong Kong automatically outputs a 3 km CAPPI of 256 km range or 512 km range to the microcomputer once every half an hour. The output of the IS-68K radar processor at Guangzhou is controlled by an operator, and 3 km CAPPI images can be sent out to the microcomputer on command.

3. COMMUNICATION PROTOCOL AND COMMANDS

3.1 Communication protocols

The communication protocol described below is used in the exchange of digital radar images and related information between Guangzhou and Hong Kong over a full duplex asynchronous circuit operating at 2 400 bps. It was jointly designed by the Royal Observatory and the Guangdong Meteorological Bureau.

Data exchange between the two ends of the telecommunication circuits is file-based. Four fundamental sequences underline the transfer of a data file. They are explained below.

(a) Transfer set up sequence

Either microcomputer can initiate a transfer, and the transfer can be in either direction. Hereafter, the microcomputer which initiates the transfer shall be called the originating station and the other one the responding station. The station from which the file is transferred shall be called the transmitting station and the station to which the file is transferred shall be called the receiving station. In a transaction, the originating station need not be the transmitting station.

The transfer set up sequence is made up of the following steps (refer Figure 3) :

- (i) The originating station sends a COMMAND frame :

<SYN><SOH><COMMAND><EOT>

where <COMMAND> is string of text which describes the type of transfer (refer section 3.3). It also determines which microcomputer is the transmitting station and which is the receiving station.

- (ii) If the COMMAND frame is correctly received at the responding station and <COMMAND> is recognized, the responding station will send an ACK frame :

<SYN><STX><EOT>

The set up is considered successful and the responding station proceeds to execute the data transfer sequence (refer (b) below). If however an error is found in the COMMAND frame received, the responding station will send a STATUS frame to indicate that an error has been detected :

<SYN><ETX><STATUS><EOT>

where <STATUS> is a byte which takes on one of the following values :

<STATUS> = 51 indicates an error has occurred in opening the disk file for data transfer;

<STATUS> = 52 indicates that a complete COMMAND frame cannot be received within 2 seconds, and time-out has occurred;

<STATUS> = 53 indicates that an error is detected in the COMMAND frame, or <COMMAND> is not recognized.

<STATUS> = 64 indicates termination at the request of the operator.

- (iii) If the originating station receives an ACK frame, the set up is considered successful and the station will proceed to execute the data transfer sequence. If however a STATUS frame with an error status is received, the set up is considered unsuccessful and the originating station will go back to step (a)(i) after a random delay. It should be noted that since both microcomputer can initiate the data transfer, COMMAND frames may be sent from both microcomputers at about the same time resulting in a clash. The randomness of the delay time is to minimize the chance of more clashes in re-tries.

(b) Data transfer sequence

Data transfer is effected through the following steps (refer Figure 4).

- (i) The receiving station sends a WINDOW frame to specify the range of data to receive :

<SYN><ACK><INDEX><SIZE><EOT>

where <INDEX> is a one byte binary value indicating the logical sequence number of the DATA block in the file data stream in modulo 128 (see also next paragraph). The normal block length is 256 bytes, but the last DATA block in a file can have less than 256 bytes. <SIZE> is a one byte binary value indicating the maximum number of DATA blocks the transmitting station can send in response to this WINDOW frame. <SIZE> is set to 8 in the program. If the telecommunication circuit is noisy with an error rate considerably worse than 10^{-6} , a smaller value of size should be used.

- (ii) On receiving the WINDOW frame, the transmitting station will, according to <SIZE> and <INDEX> in the WINDOW frame received, send a maximum of <SIZE> DATA frames starting from the <INDEX>th block of the file data stream. Each DATA frame has the following format :

<SYN><ETB><INDEX><LENGTH><DATA>
<CHKSUM0><CHKSUM1><EOT>

where <INDEX> is the logical sequence number of the DATA frame in modulo 128. It starts from 0 for the first DATA frame and counts up to 128, and re-starts from 0 if more than 128 DATA frames have to be transmitted. <DATA> is a block of data in the file being transferred. It contains exactly the number of bytes indicated by the byte LENGTH. The normal block length is 256 bytes which is represented by <LENGTH> = 0. A non-zero <LENGTH> value indicates that the DATA block is the last one of the file. If the last DATA block happens to contain exactly 256 bytes, then one more DATA frame should be sent by the transmitting station, with <LENGTH> = 255 and no DATA block. <CHKSUM0> and <CHKSUM1> are the least and most significant bytes of the 16-bit checksum value. The value is the unsigned summation of all bytes in the DATA block. A maximum of <SIZE> data frame will be sent in one go (refer previous paragraph). If the end of the data file is reached, less than <SIZE> data frames will be sent and LENGTH in the last DATA frame will not be zero.

- (iii) The receiving station receives the specified number of DATA frames. If the last DATA frame is not yet received after receiving <SIZE> number of DATA frames, it will go back to step (b)(i) to request more data. If the last DATA frame is received before <SIZE> number of DATA frames are received, it will send a STATUS frame to indicate the end of data transfer :

<SYN><ETX><STATUS><EOT>

where <STATUS> = 54 indicates that the last DATA frame is received correctly.

- (iv) When the transmitting station receives the STATUS frame, it understands that the file transfer is completed.

(c) Data error recovery sequence

If an error is detected during the data transfer sequence, the following sequence is performed to recover the data.

- (i) If the receiving station finds an error in a DATA frame received, it will send a BREAK frame (refer Figure 5) :

<SYN><ESC><STATUS><EOT>

where <STATUS> takes on one of the following values :

<STATUS> = 48 indicates that a checksum error is detected in one of the DATA frames received;

<STATUS> = 49 indicates that a DATA frame is received out of sequence.

The receiving station expects the transmitting station to stop sending the remaining DATA frames specified by the WINDOW frame, and it ignores all DATA frames received before a 1-second break in the incoming data stream. After that 1-second data break, the receiving station will go to step (b)(i) to re-transmit the previous WINDOW frame.

- (ii) On receiving a BREAK frame, the transmitting station will stop transmitting any more data, and go to step (b)(ii). If no WINDOW frame is received within 15 seconds, the transmitting station will send an ACK frame to request a WINDOW frame :

<SYN><STX><EOT>

(d) Termination sequence

There will be occasions when it is impossible to continue with a file transfer or when the operator at one end of the telecommunication circuit wants to terminate the current transfer. The following sequence will be executed to terminate the transaction (refer Figure 6).

- (i) Having detected an irrecoverable error, the receiving station will send a STATUS frame to indicate this :

<SYN><ETX><STATUS><EOT>

where <STATUS> takes on one of the following values :

<STATUS> = 51 indicates that an error has occurred in disk file operation;

<STATUS> = 52 indicates that no DATA frames is received for the 15-second time-out period despite 5 re-tries of sending the WINDOW frame at step (b)(i);

<STATUS> = 64 indicates termination at the request of the operator (refer Figure 6).

- (ii) Having detected an irrecoverable error, the transmitting station will send a STATUS frame to indicate this.

<SYN><ETX><STATUS><EOT>

where <STATUS> takes on one of the following values :

<STATUS> = 50 indicates that the WINDOW frame received requests for data beyond the end of the source data file;

<STATUS> = 51 indicates that an error has occurred in disk file operation;

<STATUS> = 52 indicates that no WINDOW frame is received for the 15-second time-out period despite 5 re-tries of sending the ACK frame from step (b)(ii).

<STATUS> = 64 indicates that termination at the request of the operator (refer Figure 6).

3.2 Time-outs and re-tries

Time-outs and re-tries are used in the communication protocol to prevent contention and lock-out due to telecommunication circuit problems or improper operation at one end of the circuit. The values used are :

- (a) Incomplete COMMAND frame time-out : 2 seconds
- (b) No DATA frame time-out : 5 seconds
- (c) Guard time after sending a BREAK frame : 1 second
- (d) Re-tries for no DATA frame (re-transmit WINDOW frame) : 10 times (refer Figure 7)
- (e) No WINDOW frame time-out : 15 seconds
- (f) Re-tries for no WINDOW frame (re-transmit ACK frame) : 10 times (refer Figure 8)
- (g) No response to COMMAND frame time-out : 10 seconds
- (h) Re-tries for no response to COMMAND frame (re-transmit COMMAND frame) : 5 times
- (i) Delay after receiving STATUS frame which indicates that there is an error in the COMMAND frame received : random value between 0 and 5 seconds.

3.3 Commands

Four commands for the COMMAND frame have been implemented so far. They are described below.

(a) Directory command

This command is used to obtain the current directory listing (in DOS format) of the microcomputer at the other end of the telecommunication circuit. The originating station will be the receiving station. The format of the command is :

DIR SS:NAME SPECIFICATION

where SS is the location whose directory is being accessed, and is either HK for Hong Kong or GZ for Guangzhou. The NAME SPECIFICATION is a string used to specify selective directory listing. The wild card characters '*' and '?' may be used to indicate an arbitrary string or character. '/W' may also be used to specify the short DOS format for directory listing.

(b) File copy command

This command is used for file transfer set up. The originating station can be either the transmitting station or the receiving station depending on the command content. The format of the command is :

COPY SS:NAME DD

where NAME is the name of the file (in DOS format) to be transferred from the source location SS to the destination location DD. SS is either HK for Hong Kong or GZ for Guangzhou, while DD is GZ or HK.

(c) Message command

This command is used for setting up message transmission. The originating station is also the transmitting station. The format of the command is :

SHOW DD LOCMMSG

where DD is the message destination location which is either HK for Hong Kong or GZ for Guangzhou. LOCMMSG is the name of the file holding the message at the originating station.

(d) Link check command

The communication protocol and the radar image exchange software may be further developed in the future. It is desirable that each side knows which version of the radar image exchange software is being used by the other side. This command is used for identifying the version number of the radar image exchange software in use. Its format is :

VER DD VERSION

where DD is the location receiving this command, and VERSION is a line of text identifying the version number of the radar image exchange software running at the originating station. The responding station should reply with a CHECK frame specifying the version number of the radar image exchange program that it is running :

<SYN><CAN>VERSION<EOT>

4. SOFTWARE FUNCTIONAL DESCRIPTIONS

4.1 Overview

The following functions are implemented in the microcomputer at each end of the telecommunication circuit :

- (a) Automatic reception of radar images from the local radar processor.
- (b) Scheduled automatic decoding, transmission and archival of radar images received from the local radar processor.
- (c) Automatic reception, decoding, display and archival of radar images received from the other end of the telecommunication circuit.
- (d) Search, decoding and display of radar images stored on disk.
- (e) Listing the current disk directory in DOS format.
- (f) Request for, reception and display of a listing of the current directory of the disk at the other end of the telecommunication circuit in DOS format.
- (g) Sending a disk file to the other end of the telecommunication circuit. The file need not be a radar image.
- (h) Request for and reception of a disk file from the other end of the telecommunication circuit. The file need not be a radar image.
- (i) Editing and transmission of an operator message to the other end of the telecommunication circuit
- (j) Automatic reception of operator messages from the other end of the telecommunication circuit.
- (k) Display of the latest operator message received as well as a log of previous messages.
- (l) Checking the telecommunication link and displaying the version number of the radar image exchange program running at the other end of the telecommunication circuit.
- (m) Logging of all transaction over the telecommunication circuit.
- (n) Displaying the status of execution of communication tasks.

Although the disk operating system in the microcomputer is essentially a single-tasking one, interrupts are used to effect simultaneous execution of more than one function. This eliminates the need to sequence operations and thus makes the system much more easy to operate.

In executing functions, status messages are displayed on the screen to enable the operator to follow the state of execution. For manual operations, screen menus are provided for ease of operation. Operation procedures are described in Chapter 6.

The radar image exchange program which runs in the microcomputer on either side essentially performs four main tasks, namely reception of images from the local radar system,

data transfer to or from the other end of the telecommunication circuit, image display and console interaction. Although the program is executed under a single-tasking operating system, it is designed to allow multiple tasks to execute more or less concurrently. This is achieved by dividing each task into small sub-tasks. The main program invokes each task in turn in a round robin manner. Depending on the status of the task invoked, a sub-task of it is executed and then program control is passed back to the main program. The time required for completing the execution of each sub-task is made small enough that no task has to wait too long for its turn to execute. Thus reception of images from the local radar system, data transfer to or from the other end of the telecommunication circuit, image display and console interaction can be carried out more or less simultaneously.

In the case of the data transfer task, it is divided into ten sub-tasks each requiring on the average less than half a second to complete. When a transfer task is invoked by the main program, the next sub-task to run is executed, and then program control is passed back to the main program. The ten transfer sub-tasks are :

- | | | |
|-----|-----------------|--|
| (a) | NOCOMMAND | ; wait for remote command |
| (b) | RECEIVECOMMAND | ; receive and decode remote command |
| (c) | TRANSMITCOMMAND | ; initiate active transfer task |
| (d) | DELAYCOMMAND | ; random delay after command collision |
| (e) | WAITCOMMANDACK | ; wait for command acknowledgement |
| (f) | WAITDATATXACK | ; wait for data acknowledgement |
| (g) | TRANSMITDATA | ; transmit the specified data |
| (h) | WAITDATA | ; wait for data |
| (i) | RECEIVEDATA | ; receive and extract data |
| (j) | RECEIVEBREAK | ; synchronization after error detected |

During data transfer, the execution of sub-tasks follows a logical sequence. Depending how the transfer task is set up, the sequence has four variations. Figure 9 shows the four possible variations of the data transfer sequence.

As described in the previous chapter, data exchange between the two ends of the telecommunication circuits is file-based. A transfer can be initiated by either microcomputer. The microcomputer which initiates a transfer is said to be performing an active data transfer while the opposite microcomputer is said to be performing a passive transfer. Furthermore, the transfer can be set up for either direction. Thus, the originating microcomputer is said to be performing an active data-out or active data-in task for transmitting or receiving a data file respectively. The opposite microcomputer is said to be performing a passive data-in or passive data-out task for receiving or transmitting the file respectively.

To initiate a transfer, the originating microcomputer will first send a COMMAND frame to the responding microcomputer and wait for reply. Since either microcomputer can initiate a transfer, a COMMAND frame can be sent from both microcomputers at about the same time resulting in a clash. The contention is resolved by each microcomputer transferring the current transfer task to a 'delayed-task' queue. The delayed task is recalled after a random delay of 0 to 5 seconds. The random seeds used in the two microcomputer are different so the delayed tasks at the two microcomputers are recalled at different times, and the one recalled first can proceed to completion. The delayed task in the other microcomputer will be recalled after the current transfer task has finished.

If both ends of the telecommunication circuit adheres strictly to the agreed radar image exchange schedule, the telecommunication channel should be free for transmission of an image at the scheduled time slot for either microcomputer. The need for both sides to adhere strictly to the agreed schedule is considered an unacceptable limitation. Furthermore, it is not always possible to stick to the schedule because manual file transfer operations are allowed. If this limitation is lifted, automatic transmission of an image by a microcomputer within the scheduled time slot may not be able to proceed because the telecommunication channel is already occupied by another transfer operation which has not yet been completed. The radar image

exchange program handles this by transferring the blocked transfer task to a 'queued-task' queue. This queued task will be recalled once the current transfer task has finished. The queue has only one position. Any new transfer task put on the queue will replace the task which is queuing there.

4.2 Automatic transmission of radar images

The flow of radar image data inside the microcomputer at Hong Kong for automatic radar image exchange is shown in Figure 10, while that for the microcomputer at Guangzhou is shown in Figure 11. The two diagrams are symmetrical, and the following description applies to both of them.

The serial port COM1 is connected to the local radar processor. There is no hardware or software handshaking control on the data coming through this port. Using an interrupt routine, any data arriving at this port will be captured and stored into a file named LOCAL.RAW in the current directory. Each radar image is assumed to be a continuous data stream with no break more than 5 seconds. A break of 5 seconds is therefore used to determine the end of a radar image file. During reception of images from the local radar system, a blinking red label 'LOCAL RECVING' is displayed at the lower right hand corner of the screen.

After a complete radar image is received at COM1, the file LOCAL.RAW is decoded to determine the image type (refer Appendix A). If it is a valid image type, it is copied to one of the specially named display files according to the image type. Each of these display files is used to store the latest radar image of a type for quick image retrieval and display. The radar image is further decoded, decompressed and display on the screen if the local radar image display is currently selected (refer section 4.4). If LOCAL.RAW is found not to have a valid image format when it is first decoded, no further processing of it will be done.

The following file names have been reserved for storing the latest radar images of the Hong Kong radar system :

File name	Type no.	
HKTEST	1	; test pattern
HKCAPLO	2	; low resolution 3 km CAPPI
HKECHOTO	3	; low resolution echo top map
HKETPPI	4	; echo top map
HKACCUMA	5	; accumulation map
HKVERTCX	6	; vortex track
HKPTCAST	7	; point forecast
HKCAP256	8	; 3 km CAPPI at 256 km range
HKCAP512	9	; 3 km CAPPI at 512 km range

Display decoders for only types 8 and 9 (3 km CAPPI at 256 km and 512 km range) are included in the radar image exchange program.

The following file names have been reserved for radar images of the Guangzhou radar system :

File name	Type no.	
GZPPI	1	; PPI
GZRHI	2	; RHI
GZCAPPI	3	; CAPPI
GZETPPI	4	; echo top PPI
GZTCW	5	; total column water content
GZZPPI	6	; elevated PPI
GZPRE	7	; precipitation

GZLOCAPP	8	; sectored CAPPI
GZVCS	9	; vertical cross-section
GZTRACK	10	; vortex track

Display decoders for all image types have been written by the Guangdong Meteorological Bureau and provided to the Royal Observatory. All of them are included in the radar image exchange program.

After updating the appropriate display file, the radar image exchange schedule is checked to see if the image needs to be transmitted to the other end of the telecommunication circuit. Before a transmission is initiated, the image is first copied to a file whose name is uniquely determined by the date, time and type of the image. The format of the file name is YYMDDhmm.STT where

YY	year code (00 to 99)
M	month (1, 2..9, A, B and C)
DD	date (01 to 31)
h	hour code (0 to 9, A to N)
mm	minute (00 to 59)
S	location ('H' or 'G' for Hong Kong or Guangzhou)
TT	image type number (00 to 10)

4.3 Reception of files

When a file, whether a radar image file or any other type of file, is sent across the telecommunication circuit, the originating station always specify to the responding station the name of the file to be transmitted. The data file received at COM2 is stored in a file of that name. If its name conforms to the naming convention for a radar image file (i.e. YYMDD-hmm.STT), it is decoded and copied to a display file of the appropriate type. The display file is further decoded and displayed on the screen if the remote radar image display is currently selected (refer section 4.4). If the file name does not conform to that of an radar image or if the data format does not conform to that of a valid image, no further processing is done. The above applies to automatic exchange of radar images as well as manual transfer of files under operator control.

4.4 Image display

The microcomputer can display only one radar image at a time. The operator can select to display either local or remote radar images at any time. Once selected, the display will be updated automatically when a corresponding radar image is received. The operator can also select and display old images by entering the file name.

4.5 Directory listing

The operator can command the local microcomputer to display a listing of the current directory. The listing is first obtained through a DOS system call, then it is stored in the file LOCDIR1 and displayed on the screen in DOS format.

The operator can also request a listing of the current directory of the microcomputer at the other end of the telecommunication circuit. On receiving such a request, the responding microcomputer will make a system call to obtain a listing of the current directory. The listing is stored in a file called LOCDIR which is then transmitted. After receiving the directory listing, the requesting microcomputer will store it in the file REMDIR, which can be displayed on the screen by the operator.

4.6 Manual file transfer

Under operator control, a file can be transferred from one end of the telecommunication circuit to the other. The operator will have to enter the name of the file to be transferred and the direction of the transfer. If the file exists at the transmitting end, and no file of that name exists at the receiving end, the file will be transferred; otherwise, a message will be displayed to tell the operator that a file access error has occurred. A file received will be treated as in section 4.3.

4.7 Dialogue

Text messages can be exchanged between the microcomputers at the two end of the telecommunication circuit under operator control.

Message reception is automatic. When a message is received from the other end of the telecommunication circuit, it is placed in a file REMMSG. A blinking red label 'MESSAGE IN' will be displayed at the lower right hand corner of the screen and the microcomputer's loudspeaker will be sounded once every minute. These visual and audio signals will continue until the message is read by an operator whereupon the message is appended to the end of a message log file MSGLOG for future reference. The content of the REMMSG is then cleared. If a new message is received before the last received message is read, the old message will be replaced by the new one.

To prepare a message for transmission, the operator can use a built-in message editor which has some primitive editing functions such as cursor movement, insert and delete. The message being edited is stored in the file LOCMSG. A message which is transmitted is also appended to the same message log file MSGLOG for future reference.

For each message appended to the message log file MSGLOG, the source location (in this case either Hong Kong or Guangzhou) and the date and time of reception or transmission are added to the beginning of the message. Within the radar image exchange program, the operator can examine the content of this file but cannot modify it.

5. PROGRAM MODULE DESCRIPTION

5.1 Overview

The radar image exchange software is made up of the following components :

- (a) MODE.COM ; DOS command for setting the baud rate and byte format of communication ports COM1 and COM2
- (b) LCOM.EXE ; software driver for COM1 and COM2
- (c) DRIEP.H ; header file for MAIN.C defining station names
- (d) CONSOLE.H ; header file for CONSOLE.C defining screen layouts
- (e) DECODE.H ; header file for DECODE.C defining radar image types
- (f) RPROTO.H ; header file for RPROTO.C defining the communication protocol
- (g) MAIN.C ; main program loop
- (h) COMM.C ; module for handling data to and from COM1 and COM2
- (i) CONSOLE.C ; module for handling keyboard input and screen output
- (j) DECODE.C ; module for radar image decoding and flow control
- (k) GZRADAR1.C ; module for decoding Guangzhou radar images
- (l) COLOR1.ASM ; module for fast pixel plotting on the screen
- (m) LPROTO.C ; module for reception of radar images from local radar
- (n) RPROTO.C ; data exchange module
- (o) VIEW.C ; file browsing module.

Modules (a) and (b) are commercial products. Modules (k) and (l) contains C and assembly codes developed by the Guangdong Meteorological Bureau for decoding and displaying radar images of the Guangzhou radar system. The rest of the software is developed by the Royal Observatory.

The software for the two ends of the telecommunication circuit are nearly identical. The only difference is in the order of station names, which is defined in module DRIEP.H.

5.2 Program module description

A functional description of the program modules in the radar image exchange software is given below. Detailed program flow description is omitted for brevity.

(a) LCOM.EXE

This commercial utility program is used to provide the most basic I/O functions for communication ports COM1 and COM2. All these functions are accessed by a software interrupt vector. The following functions are used in the radar exchange program :

Function no.	Function description
1	Close a COM port
2	Write data to a COM port
3	Read data from a COM port
4	Set COM port options (baud rate, parity, etc.)
5	Return state of COM port options
6	Special functions
9	Open and initialize a COM port

LCOM.EXE must be loaded before the radar image exchange program is executed.

Each function of LCOM.EXE is invoked by generating a software interrupt vector 067H with the corresponding parameters loaded into the microprocessor's registers. Details are given in Reference 5.

(b) MAIN.C

This module is the main program. The main program starts by invoking the initialization codes of other modules. Then in a round robin manner, it invokes each task in turn (refer section 4.1) by making a call to the task entry point. One sub-task of the task will be executed and then program control is passed back to the main program.

(c) COMM.C

This module is an interface to the resident utility program LCOM.EXE. It is used for setting up two data buffers for COM1 and COM2, reading their status, and reading data from and writing data to them. Each of these two data buffers is further divided into an input buffer and an output buffer.

Data coming into a communication port are placed in the input buffer by LCOM.EXE, and must be read frequently enough by this module to avoid data overflow. Data to be output through the communication port are transferred by this module to the output buffer. The actual output will be done by LCOM.EXE

The following functions and procedures are include in this module :

(i) LCOMinit()

It sets up the COM ports as follows :

COM	Interrupt	Address	Buffer size (In)	Buffer size (Out)
1	4	3F8H	4000	200
2	5	2F8H	1500	3000

(ii) OutputCOM(port,OutputStr)

It transfers a null-terminated string to a COM port output buffer.

(iii) DrainInCOM(port)

It flushes the input buffer of the COM port.

(iv) DrainOutCOM(port)

It flushes the output buffer of the COM port. This function is especially useful when the transmitting station receives a BREAK frame and has to stop further transmission of remaining DATA frames in the window.

(v) QueueSizeCOM(port)

This function returns the data queue size of the COM input buffer.

(vi) ReadStrCOM(port,ComMsg)

It transfers data from the COM port input buffer to a structured data buffer. The structured data buffer allows null bytes to be transferred.

(vii) WriteChrCOM(port,chr)

It transfers a character to a COM port output buffer.

(viii) WriteStrCOM(port,ComMsg)

It transfers data from a structured data buffer to a COM port output buffer. The structured data buffer allows null bytes to be transferred.

(ix) CloseCOM(port)

This function closes a specified COM port for interrupt processing.

(x) OpenCOM(port,Intr,Adrs,buffer,InSize,OutSize)

This function opens a COM port, initializing its hardware for interrupt processing.

(d) CONSOLE.C

This module handles keyboard input and screen output. The following functions and procedures are included in this module :

(i) TaskInit()

Initialization for screen output.

(ii) InitKeyboard()

Initialization for keyboard operation.

(iii) InitScreen()

It sets up the screen layout.

(iv) SetupScreen()

It initializes the screen to 640 pixels x 480 lines x 16 colours graphics mode.

- (v) `DisplaySystemTime()`
It updates the date/time stamp and sets time related task flags.
- (vi) `ClearMenuScreen()`
It clears the screen area for displaying a new menu.
- (vii) `CheckKeyboard()`
It determines the sub-task which is to be executed next.
- (viii) `CheckEditMsg()`
It handles the keyboard input for the message editor.
- (ix) `DisplayMessage()`
It handles the screen output for the message editor.
- (x) `CheckViewPage()`
It handles the keyboard input for text file view function.
- (xi) `CheckConfirm()`
It handles the keyboard input for manual operation confirmation.
- (xii) `PopMenu()`
It handles the screen output for menu escape.
- (xiii) `EditFileSpec()`
It handles the keyboard input for file specification editing.
- (xiv) `GetKey()`
It is a function which reads scan codes from the keyboard.
- (xv) `kbchar()`
It is a function which reads a character from the keyboard.
- (xvi) `CheckMenu0()`, `CheckMenu1()`, `CheckMenu2()`, `CheckMenu3()`,
`CheckMenu4()`
They handles keyboard input for menu operation.
- (xvii) `SetupStatusScreen()`
It changes the screen to display communication task status.
- (xviii) `DisplayImageMenu(station)`
It displays the image menu for the specified station.

(xix) `FileDate(FileSpec)`

It is a function which returns the date string from the given radar image file name.

(xx) `FileTime(FileSpec)`

It is a function which returns the time string from the given radar image file name.

(xxi) `FileTypeLabel(FileSpec)`

It is a function which returns the radar type from the given radar image file name.

(xxii) `DisplayMenu()`

It displays the previously selected menu.

(xxiii) `DisplayMenuBox()`

It displays the menu layout.

(xxiv) `CtrlChandler()`

It is the Ctrl-C keyboard interrupt handler.

(xxv) `DisplayMsgCursor()`

It displays the cursor for the message editor.

(xxvi) `DisplayStatus()`

It displays the communication status and system message.

(e) `DECODE.C`

This module contains the decoders for radar images of the Hong Kong radar system. It also schedules radar image exchange, and assign names to local radar images. The actual decoders for the Guangzhou radar system are in a separate program module.

The following functions and procedures are include in this module :

(i) `DecodeLocalRadarImage(filename)`

It invokes procedures to process a recently received local radar image.

(ii) `DecodeRemoteRadarImage(filename)`

It invokes procedures to process a recently received remote radar image.

(iii) `MakeFileName(WorkStn,PictureType,InStr,OutStr)`

It makes up an image file name (DOS format) using the supplied parameters.

(iv) DecodeHK(InputFile,DateStr)

It decodes the specified Hong Kong radar image file for screen display and returns its date string. This decoding task may take up to 7 seconds to complete.

(v) DecodeGZ(InputFile,DateStr)

It invokes the decoder (in a separate program module) for the specified Guangzhou radar image file for screen display and returns its date string.

(vi) CheckDecodeLocalRx()

It checks if a local radar image has to be processed.

(vii) CheckDecodeRemoteRx()

It checks if a remote radar image has to be processed.

(viii) CheckOutputSchedule()

It checks the pre-programmed schedule to see if a recently received local radar image has to be transmitted automatically to the other end of the telecommunication circuit.

(f) LPROTO.C

This module handles the reception of images from the local radar system. The following functions and procedures are included in this module :

(i) LocalRxInit()

Local radar image reception task initialization.

(ii) CheckLocalRx()

It is the local radar image reception task entry point.

(g) RPROTO.C

This module handles data exchange with the other end of the telecommunication circuit. The following functions and procedures are included in this module :

(i) KillCurrentRemoteTask()

It stops the current data transfer task.

(ii) KillDelayedRemoteTask()

It cancels the delayed data transfer task.

(iii) KillQueuedRemoteTask()

It cancels the queuing data transfer task.

(iv) SetXchgTaskInQ(CmdFmtNum,FileName)

It assigns the not-yet-started current task to the 'queued-task' queue.

(v) StartXchgTask(CmdFmtNum,FileName)

It activates the specified data transfer task.

(vi) SetTaskCommand(CmdFmtNum,Task,FileName)

It sets up the COMMAND according to the specified parameters.

(vii) OpenXchgFile()

It opens the file required for the current data transfer task.

(viii) CheckSum(MsgPtr,Length)

It is a function which calculates the checksum of the specified data block.

(ix) RemoteRxTxInit()

Initialization for data exchange control.

(x) InitTaskParm()

Initialization required for starting a new data transfer task.

(xi) CheckLastCode(Message,Code)

It checks the control codes at the end of a frame in the data buffer.

(xii) StopXchgTask()

It is the error termination of a data transfer task.

(xiii) CloseXchgFile()

It is a function which closes the file used in the last data transfer task.

(xiv) EndXchgTask()

It is the normal termination of a data transfer task.

(xv) IsDateNamedFile(Name)

It is a function which checks the validity of the given radar image file name.

(xvi) RecallXchgTask(Task)

It recalls a delayed task or queued task.

(xvii) CheckRemoteRxTx()

It determines which sub-task of the data transfer task is to be executed next.

(xviii) ChkBlkNum()

It is a function which checks the validity of the current block sequence during reception of data from the other end of the telecommunication circuit.

(xix) CheckWAITDATATXACKretry()

It is a function which keeps track of re-tries for no DATA frame received during reception of data from the other end of the telecommunication circuit.

(xx) CheckCommandRetry()

It is a function which keeps track of re-tries for error in command response during data transfer set up.

(xxi) ConnectString(to,from,count)

It is a function which connects two structured data buffer. The data inside these buffers may contains null bytes.

(h) VIEW.C

This module contains file browsing subroutines for the CONSOLE.C module. The following functions and procedures are included :

(i) ClearViewWindow()

It clears the screen for displaying new data.

(ii) ViewNextPage(view_fp)

It displays the next page of data.

(iii) ViewPrevPage(view_fp)

It displays the previous page of data.

(iv) ViewPrevLinePage(view_fp)

It displays one page of data starting from the last line of the previous page.

(v) ViewNextLinePage(view_fp)

It displays one page of data starting from the second line of the current page.

(vi) ViewFirstPage(view_fp)

It displays the first page of data.

(vii) ViewLastPage(view_fp)

It displays the last page of data.

(viii) EraseToLineEnd()

It clears the screen from the current cursor position to end of the line.

(ix) EraseToScreenEnd()

It clears the screen from the current cursor position to end of the current display window.

(i) GZRADAR1.C

This module contains image decoders for Guangzhou radar images. The source codes are developed and provided to the Royal Observatory by the Guangdong Meteorological Bureau. The following functions and procedures are included in this module :

(i) DecodeGZradar(filename)

Main decode program for the specified Guangzhou radar image file.

(ii) ppdisplay(), vcsdisplay(), locappdisplay()

Decoders for different types of radar images. Decoding an image may take up to 7 seconds.

(iii) pixel(x,y,colorcode)

It is a function which plots one pixel on the screen. It results in the content of the VGA display adaptor's internal registers being modified.

(iv) closegraph()

It restores the default contents of the VGA display adaptor's internal registers after plotting by the pixel(x,y,colorcode) function.

(v) initppi(range), initvcs(), initlocappi()

It generates the screen layout for different radar image types.

(vi) drawlandmark(x1,y1,x2,y2,range)

It displays the locations of Hong Kong and Macao as crosses on a CAPPI type radar image. The following coordinates are used :

Radar antenna at Guangzhou : lat = 23°07'55" N
long= 113°17'55" E

Royal Observatory, Hong Kong : lat = 22°18'13" N
long= 114°10'19" E

Distance and bearing from
Guangzhou radar antenna : 128.4 km, 135.6°

Geophysical Observatory, Macao : lat = 22°07'48" N
long= 113°21'36" E

Distance and bearing from
Guangzhou radar antenna : 111.5 km, 176.8°.

(vii) printA()

It displays the colour look-up table.

(viii) `disptype()`

It displays the type numbers for different types of radar images.

(ix) `ppiparam()`, `vcsparam()`, `rhiparam()`, `locparam()`, `trackparam()`

It displays the names and measurement units for different types of radar images.

(x) `crossword(x,y,color)`

It is a function which draws a cross in a TRACK type image display.

(xi) `convtdata(dig)`

It is a function which returns a value which is the reverse of the two bytes in a 16-bit word.

(j) `COLOR1.ASM`

It is an assembly language subroutine for fast pixel plotting called by `pixel(x,y,colorcode)` which was developed by the Guangdong Meteorological Bureau.

5.3 Program call tree

Appendix B shows the program call tree. Starting from first column, the program module 'main' invokes the initialization codes in other modules and then invokes the main tasks in the second column. The third column onward show the functions and procedures called by each of these tasks. The symbol '...' means that the function or procedure is re-used.

5.4 Program compilation and linking

To facilitate compilation and linking of the software, two MAKE files `WSHK.MAK` and `WSGZ.MAK` have been written. The former is used for compiling and linking the radar image exchange program for Hong Kong, and the latter for Guangzhou. The procedure for compiling and linking the software is as follows.

- (i) Assemble `COLOR1.ASM` to obtain the object module `COLOR1.OBJ`.
- (ii) Change definition in `DRIEP.H` so that it contains `#define WKSTN HK` for the program to be used in Hong Kong, or `#define WKSTN GZ` for the program to be used in Guangzhou.
- (c) Execute `MAKE WSHK.MAK` to generate `WSHK.EXE` for use in Hong Kong or execute `MAKE WSGZ.MAK` to generate `WSGZ.EXE` for use in Guangzhou.

5.5 Program limitation

Two colour palettes of the VGA display adaptor are used for displaying images from the two radar systems. The palette is change automatically according to the source of the image (i.e. whether it is a Hong Kong radar image or a Guangzhou radar image). The subroutine which makes the palette change cannot be interrupted while it is executed. Interrupts are therefore disabled during its execution. This may cause loss of data when receiving images from the local radar system. As reception of radar images is more important than display them, palette change is inhibited during reception of data from the local radar system. If during this time the operator switches between the display of radar images from both radar systems, the radar images from one system will appear with spurious colour.

With a 80286-based microcomputer running at 10 MHz, loss of data may occur in receiving data from the local radar system during keyboard operation if the baud rate of the communication port connecting the local radar system is greater than 2 400 bps. In that case, keyboard operation should be avoided during reception of radar images from the local radar system.

6. OPERATION PROCEDURES

6.1 Overview

The operator interface is menu driven so that the operator need not remember any command names. The state of execution of each function, whether it is automatically invoked or executed by the operator, is displayed in the form of status messages so that the operator can follow its progress.

To start the radar image exchange software, three programs should be executed in sequence. DOS command MODE.COM should first be executed to set up COM1 as well as COM2 for the appropriate baud rate, parity, number of start bits, data bits and stop bits. LCOM.EXE should then be executed with the parameter -i 103 to set up the communication ports for interrupt driven I/O with the interrupt vector 00067. Finally the radar image exchange program WSHK.EXE (for Hong Kong) or WSGZ.EXE (for Guangzhou) should be executed. Operation of the radar image exchange program is explained below. The radar image exchange program can be terminated any time by pressing Ctrl-C at the keyboard.

6.2 Operator interface menu

The menu has two levels. Figure 12 shows its structure. The key for choosing any function is indicated beside it. Functions shown in italics have not yet been implemented.

(a) Image display selection

Press '1' or '2' on the MAIN MENU to select either Hong Kong radar image display or Guangzhou radar image display. The last captured image will be displayed.

Press 'space-bar' to display the IMAGE MENU. The default image type is high-lighted in red on the menu. To display other types of image, press the corresponding key ('0' to '9').

To display an old image stored in the current directory, the operator should first determine which type of image to display as described in the paragraph above. After that, press 'A' to display the 'FILE SPEC' prompt and enter the file name and extension specification for the search. Wild card characters '?' and '*' may be used. Press 'Down' to begin search of image files meeting that specification. If such files exist, the OLD FILES MENU will be displayed together with information of the first file found. The information includes the date and time when the image was acquired and the size of the image file. The order in which the files show up depends on the ordering of files in the disk directory.

The following keys can be used in the OLD FILES MENU :

Key	Function
'Up'	search for next available image file
'Down'	search for previous available image file
'Home'	search for the first available image file
'End'	search for the last available image file
'PgUp'	search back for ten available image file
'PgDn'	search forward for ten available image file
'Enter'	display the found image
'F2'	set the found image as the last captured image
'F3'	write protect the found image file
'F4'	unprotect the found image file

'Alt-F9'	delete the found image file
'Esc'	return to IMAGE MENU

(b) Manual file exchange operations

Press '3' on the MAIN MENU to enter the EXCHANGE MENU, and then press one of the following keys for the required function :

Key	Function
'1'	transmit a file
'2'	request a remote file
'3'	request a remote directory listing
'4'	view an acquired remote directory listing
'5'	view the local directory listing
'6'	kill current data transfer task
'7'	kill delayed data transfer task
'8'	kill queued data transfer task

If '1', '2', '3' or '5' is selected, the 'FILE SPEC' prompt will be displayed. Type in the file specification and then press 'Enter' to start the operation. For operations '3' and '5', the wild card characters '*' and '?' can be used in the file specification. '/w' can also be appended to specify short directory listing. The exact file name must be used in the case of operations '1' and '2'. Alternatively, instead of entering FILE SPEC, keys 'Up', 'Down', 'Home', 'End', 'Pgup' and 'Pgdn' can be used to search for matched files in the local current directory.

(c) View remote directory

Press '3' on the MAIN MENU to enter the EXCHANGE MENU. Press '3' again on the EXCHANGE MENU and the 'FILE SPEC' prompt will be displayed. Type in the file specification and press 'Enter'. Wait for the transfer of the directory listing to complete. Press '4' on the EXCHANGE MENU to view the remote directory.

The following keys can be used in the VIEW WINDOW :

Key	Function
'Up'	scroll up one line in the view window
'Down'	scroll down one line in the view window
'Home'	position the view window to the beginning of file
'End'	position the view window to the end of file
'PgUp'	display previous page of data
'PgDn'	display next available page of data
'Enter'	exit VIEW WINDOW without clearing it
'Esc'	exit VIEW WINDOW and clear it

(d) View local directory

Press '3' on the MAIN MENU to enter the EXCHANGE MENU. Press '5' on the EXCHANGE MENU and the 'FILE SPEC' prompt will be displayed. Type in the file specification and press 'Enter'. Use the cursor keys as described in the previous paragraph for the VIEW WINDOW.

(e) Transmit a file

Press '1' on the MAIN MENU to enter the EXCHANGE MENU. Press '1' on the EXCHANGE MENU and the 'FILE SPEC' prompt will be displayed. Type in the exact file name and press 'Enter'.

The file transfer operation will be aborted if the specified file does not exist in the current local disk directory or if a file with the same name already exists in the current disk directory at the other end of the telecommunication circuit.

(f) Request a file from the other end of the circuit

Press '1' on the MAIN MENU to enter the EXCHANGE MENU. Press '2' on the EXCHANGE MENU and the 'FILE SPEC' prompt will be displayed. Type in the exact file name and press 'Enter'.

The file transfer operation will be aborted if a file with the same name already exists in the current local disk directory or if the specified file does not exist in the current disk directory at the other end of the telecommunication circuit.

(g) Terminate telecommunication tasks

Functions '6', '7' and '8' will be effective only when the corresponding communication tasks have been activated, whereupon the operator is asked to confirm the selected operation. Press 'Y' to confirm or any other key to cancel. When executed, the corresponding data transfer task is aborted, and the destination file for data reception is abandoned.

(h) Dialogue operation

The system allows message-based dialogue between the two ends of the telecommunication circuit.

Press '4' on the MAIN MENU to enter the DIALOGUE MENU, then press one of the following keys for the required function :

Key	Function
'1'	prepare a new message
'2'	edit the last message
'3'	view all past messages
'4'	view the last message sent
'5'	view the last message received

To prepare a new message, press '1' on the DIALOGUE MENU to enter the EDIT WINDOW. To edit the last message, press '2' on the DIALOGUE MENU to enter the EDIT WINDOW and bring in the last message. Type in a new message or edit the old message. The following edit and cursor keys can be used :

Key	Function
'Up'	move the cursor up one line
'Down'	move the cursor down one line
'Enter'	new line
'Backspace'	erase character left of cursor
'Del'	erase character at cursor
'Esc'	exit the EDIT WINDOW

When finished, press 'F1' and then 'Y' to confirm transmission of the message.

When a message has been received, the microcomputer will sound the loud-speaker once every minute. A red label 'MESSAGE IN' will blink on the lower right hand corner of the screen. To view the message, press the 'Esc' key to return to MAIN MENU. Press '4' to select the DIALOGUE MENU and press '5' to display the message received.

To view all past messages, select DIALOGUE MENU as above and then press '3' to enter the VIEW WINDOW. The content of the file MSGLOG is displayed. Press 'Home', 'End', 'PgUp', 'PgDn', 'Up' and 'Down' to browse the file.

6.3 Examining history of telecommunication activities

All transactions on the telecommunication circuit are logged in the file LINKLOG with date and time stamp as lines of ASCII text. The file can be displayed on the screen or printed on a printer after the radar image exchange program is terminated.

6.4 Data archival

The size of radar images of the Hong Kong radar system varies from about 13 to 20 kB depending on the amount and complexity of radar echoes, while that of the Guangzhou radar system varies between about 4 and 20 kB. According to the current schedule of image exchange, 48 images of the Hong Kong radar system and 8 images of the Guangzhou radar system are exchanged and stored on disk per day. With a 32 MB hard disk, about a month's radar images can be stored. Hence the radar image files should be removed from the hard disk no less frequently than once a month. It is actually preferable to archive radar image files and clear the hard disk once a week because an excessive number of files in the hard disk will slow down file access and consequently the overall system response.

The best way to archive radar image files is to exit the radar image exchange program and perform a hard disk backup either onto floppy diskettes or cartridge tapes. After the backup, the radar image files on the hard disk can be deleted.

The files LINKLOG and MSGLOG for logging telecommunication activities and dialogue also grow with time. They should be archived and cleared regularly just like the radar image files.

7. CONCLUDING REMARKS

The radar image exchange system described in this manual was put into operation in October 1990. Its performance has been satisfactory so far. The communication protocols and commands designed for this application were also found to be efficient.

Although the system was initially designed for use between the Royal Observatory and the Guangdong Meteorological Bureau, it can be quite easily adapted for use between any other meteorological centres. Furthermore, it can also be easily extended for the exchange of additional classes of meteorological data such as satellite images.

REFERENCE

1. Patrick Sham and Li Zechun 1990 Memorandum of Understanding between the Royal Observatory, Hong Kong and the National Meteorological Centre on the Upgrading of the Hong Kong - Beijing Meteorological Telecommunication Circuit
2. AST Research Inc. 1988 AST Premium/286 User's Manual
3. AST Research Inc. 1988 Microsoft Disk Operating System
4. Microsoft Corporation 1989 Microsoft C 5.1 Language Manual (including the Programmer's Guide, the Run-time Library Reference, the User's Guide, the Mixed-language Programming Guide and the Quick C Compiler Programmer's Guide)
5. Blaise Computing Inc. 1989 Turbo Async Plus

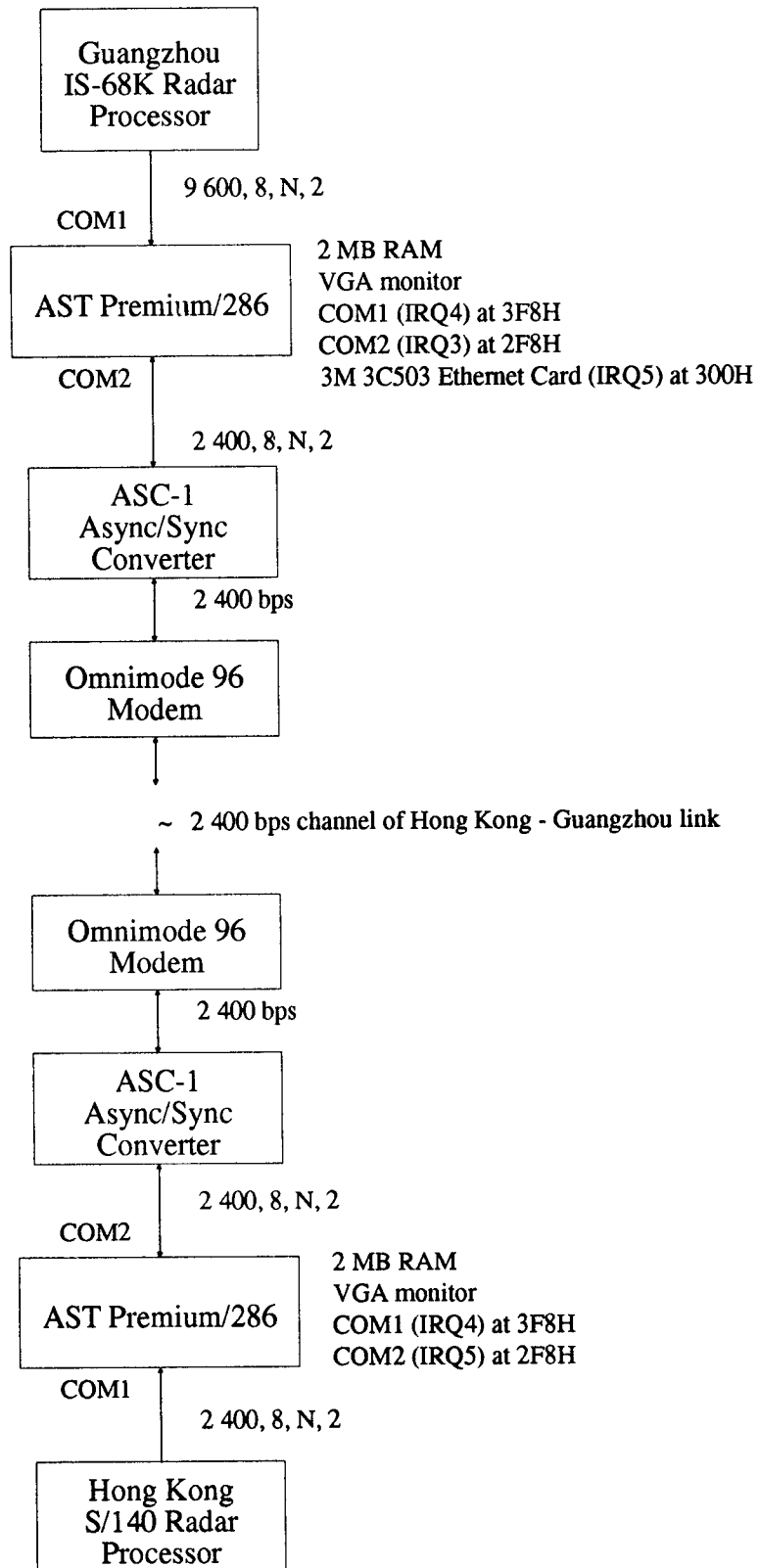
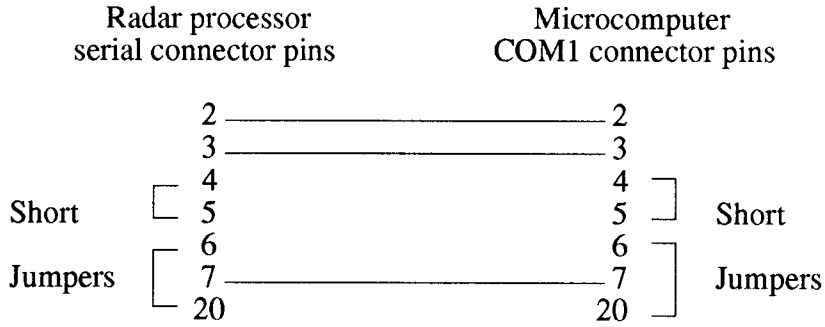


Figure 1. System block diagram

Microcomputer to radar processor connection



Microcomputer to async/sync converter connection

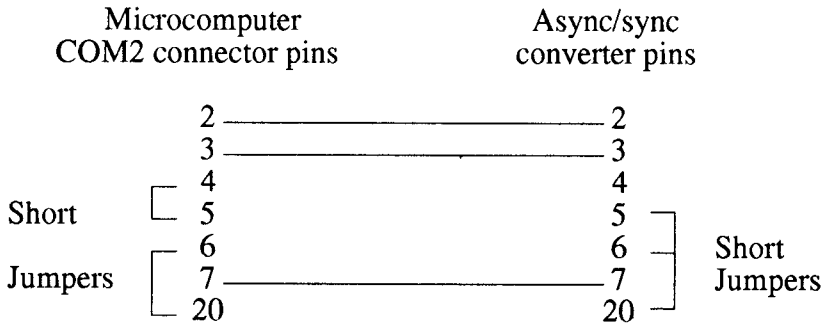


Figure 2. Cable connections to the microcomputers

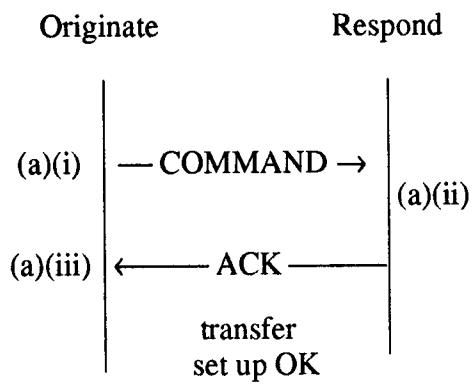


Figure 3. Transfer set up sequence

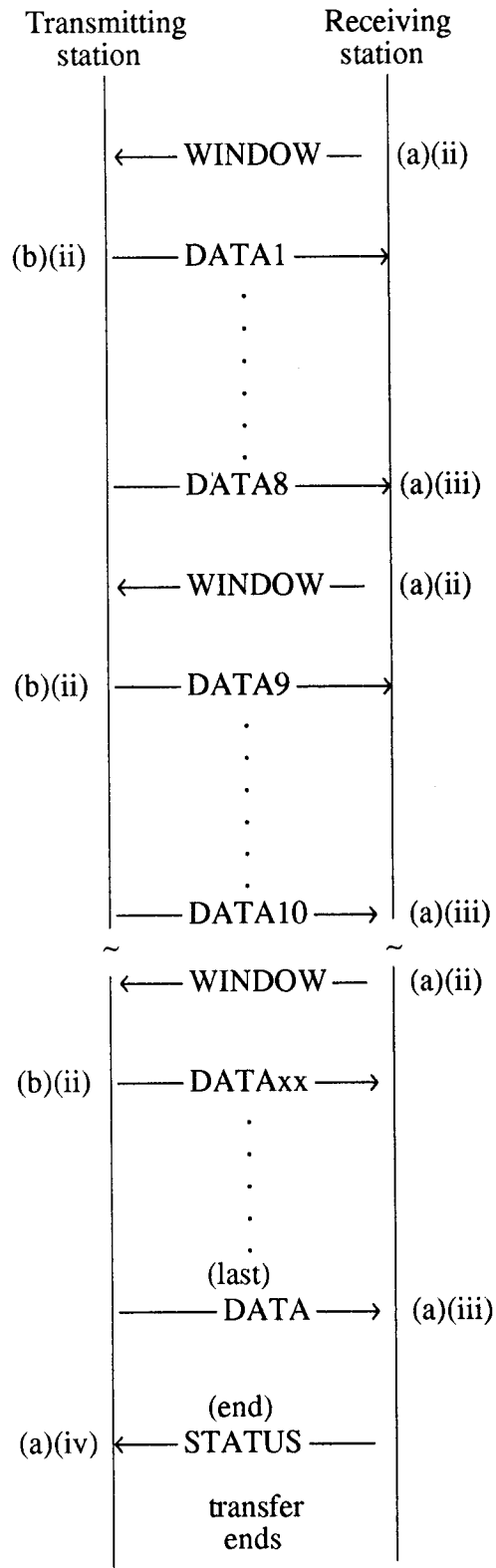


Figure 4. Data transfer sequence

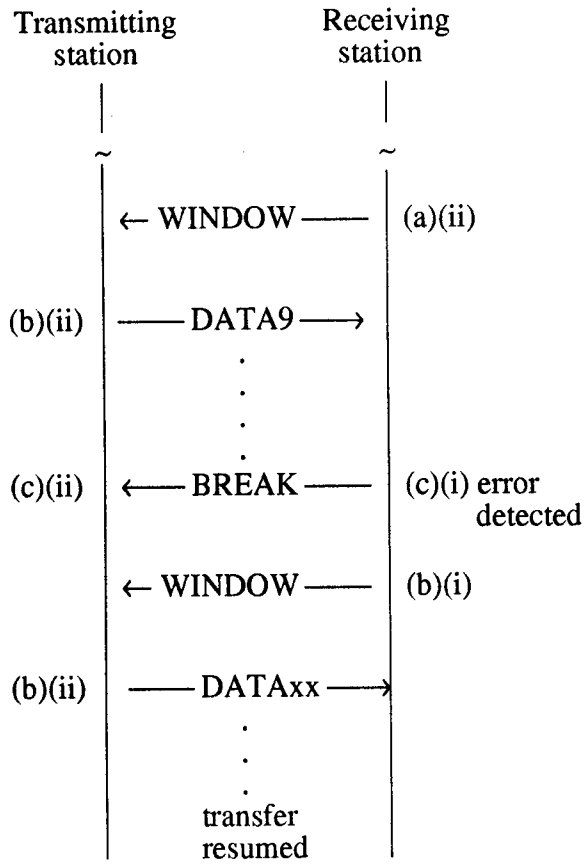


Figure 5. Data break recovery sequence

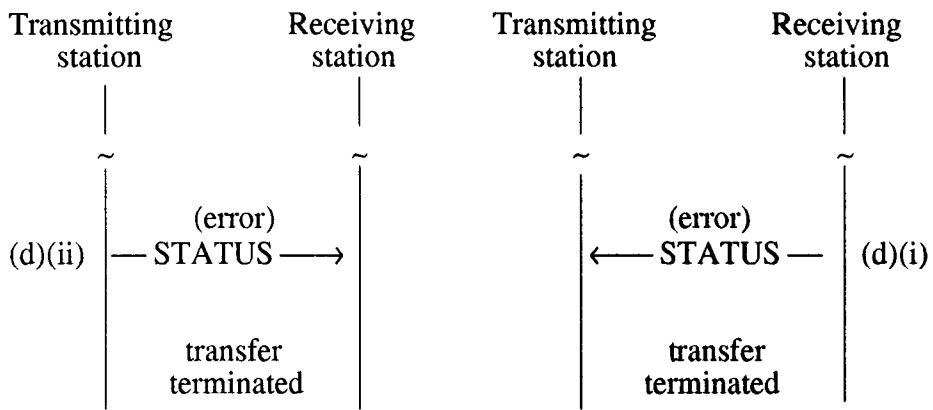


Figure 6. Termination sequence

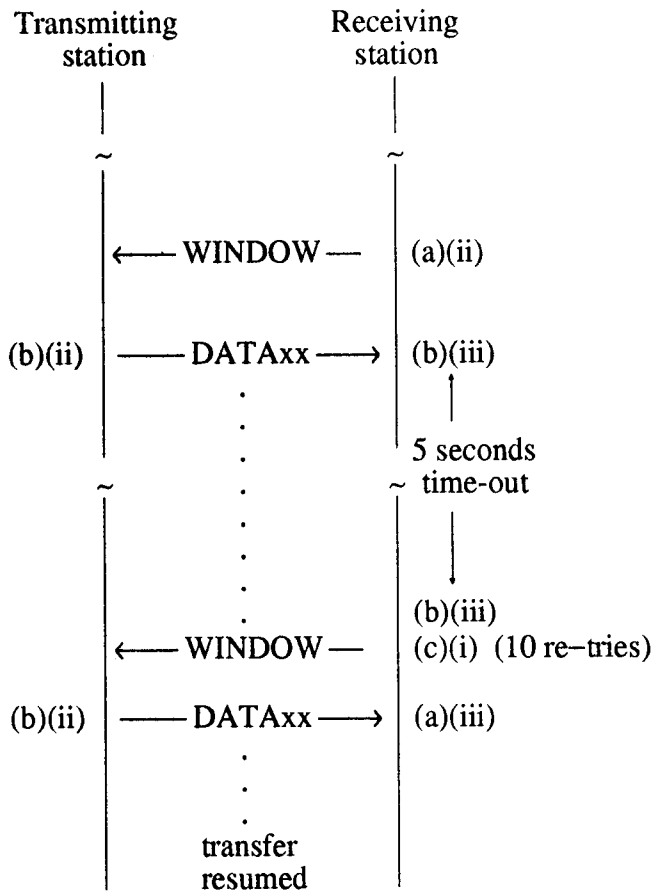


Figure 7. No DATA frame time-out recovery sequence

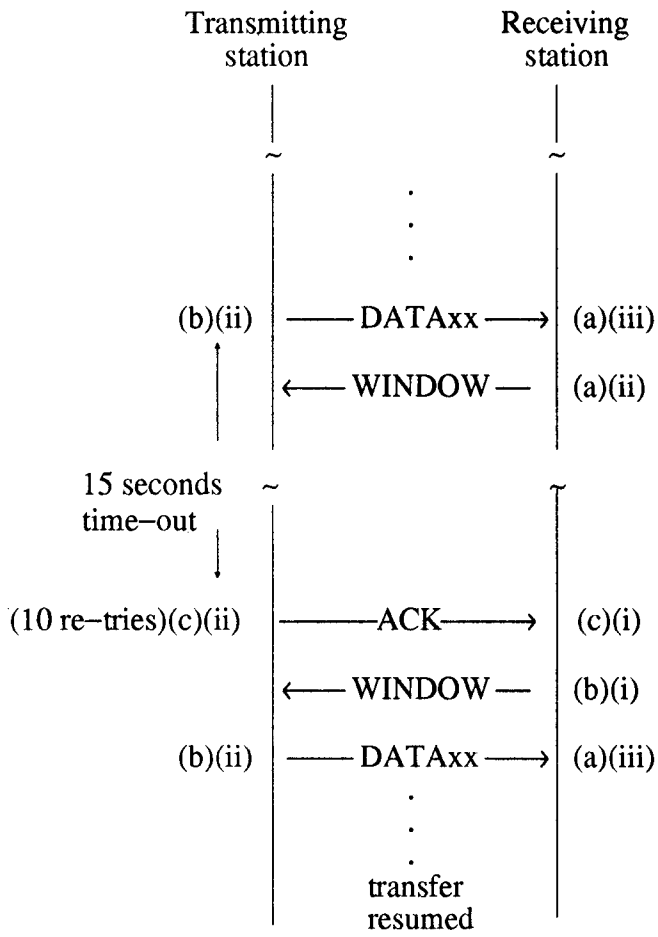
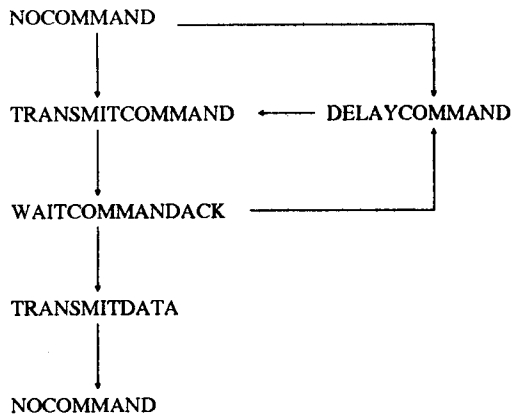
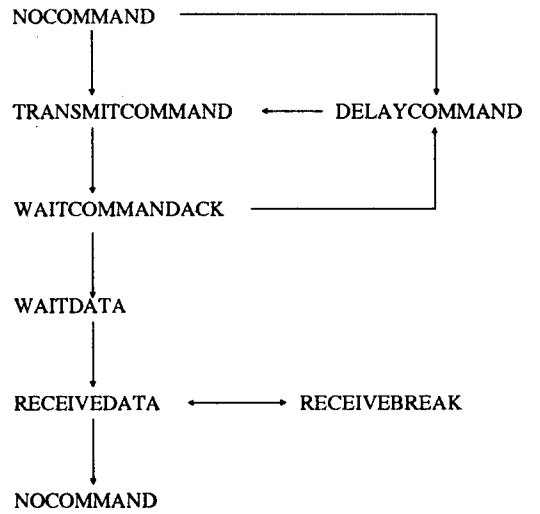


Figure 8. No WINDOW frame time-out recovery sequence

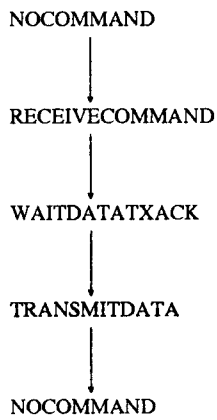
Active data-out :



Active data-in :



Passive data-out :



Passive data-in :

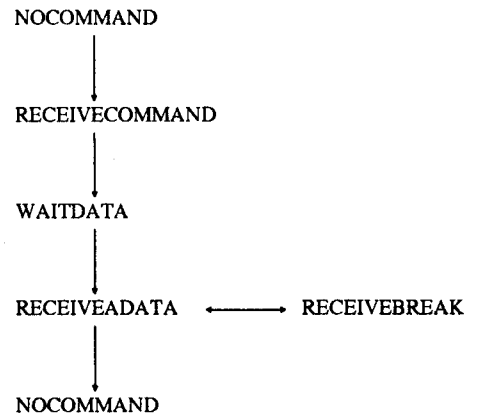


Figure 9. Operator interface menu

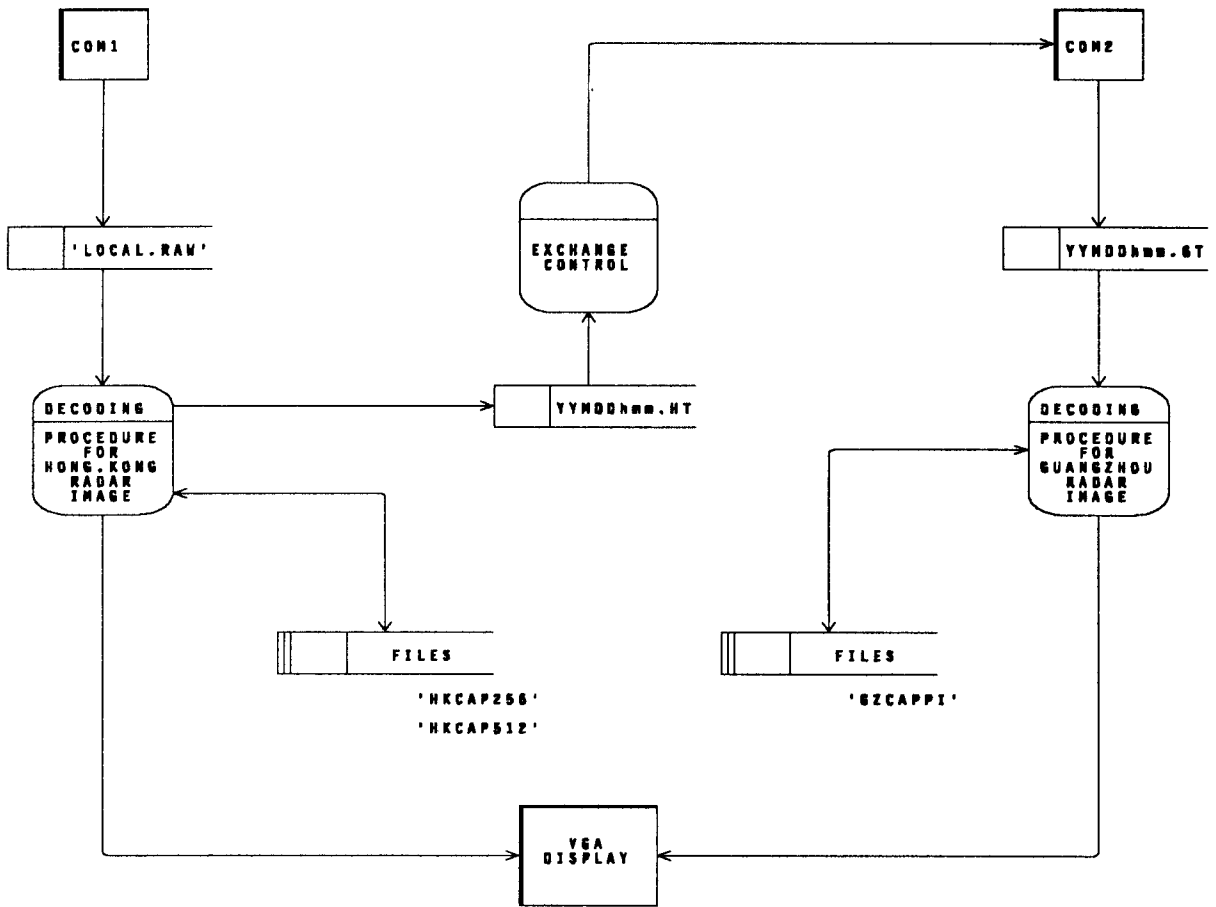


Figure 10. Data flow diagram for the microcomputer at Hong Kong

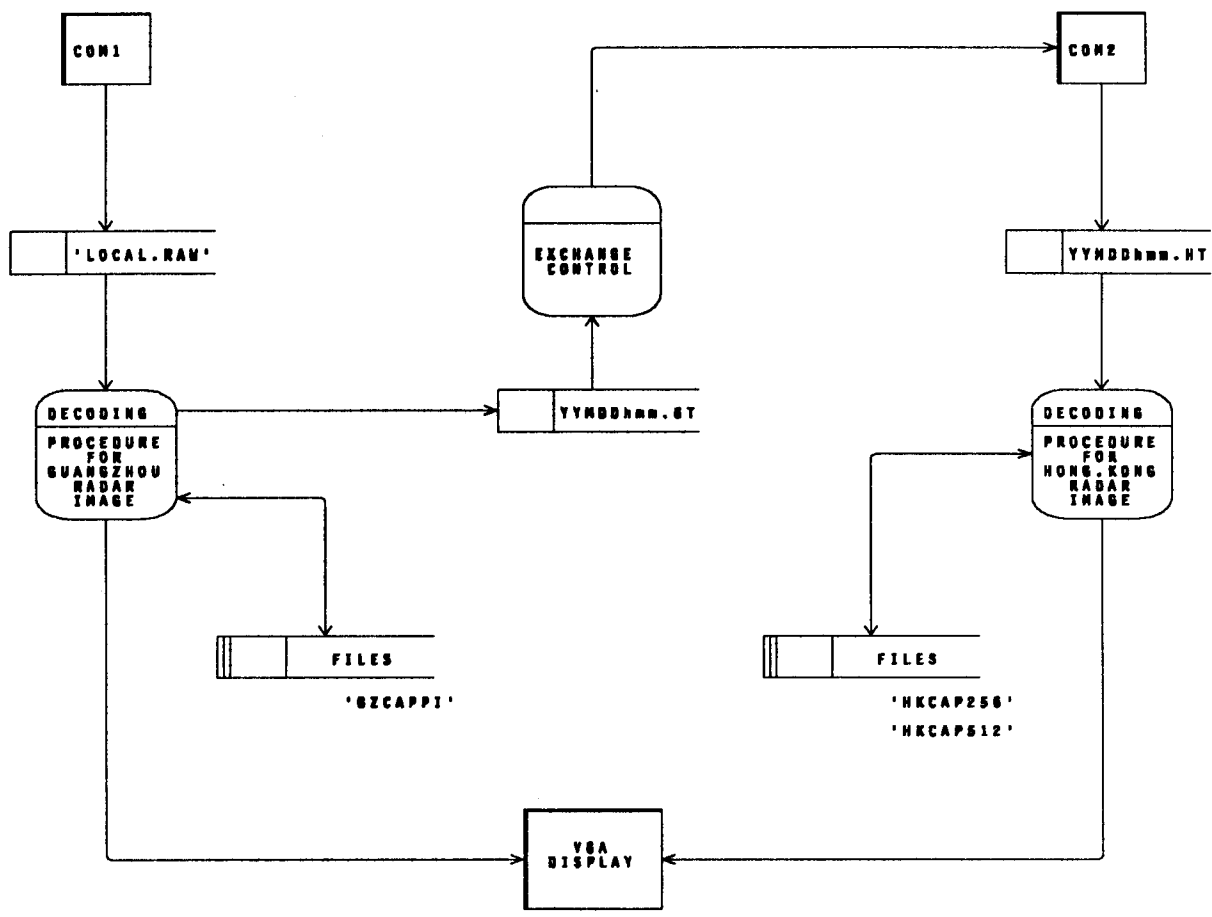


Figure 11. Data flow diagram for the microcomputer at Guangzhou

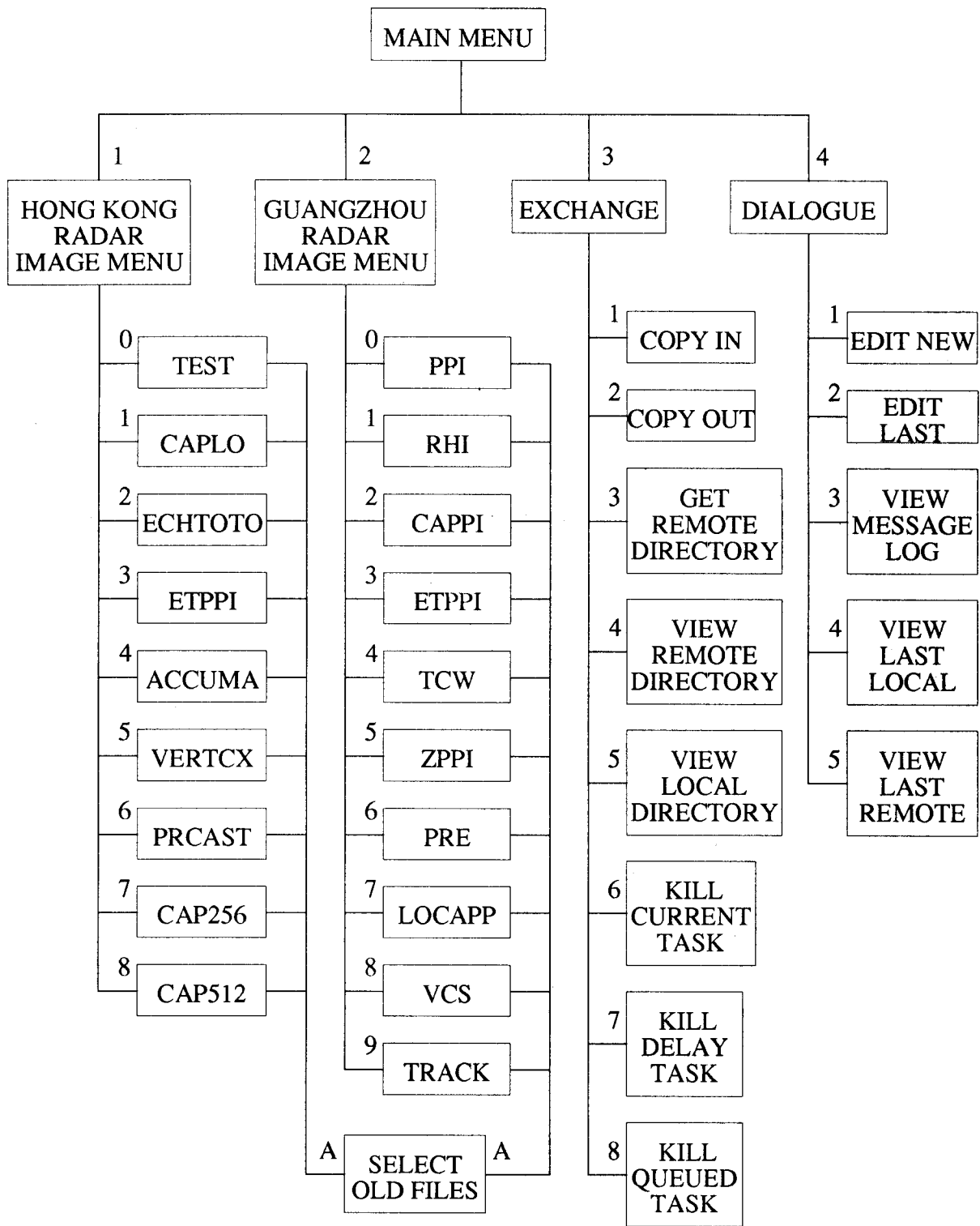


Figure 12. Operator interface menu

APPENDIX A - Radar Image Data Format

A.1 Hong Kong radar image format

Each image is made up of 256 pixels x 256 lines x 16 colours. The image file has the following sequence of data :

```

<00H><01H><PICTURE TYPE>
<00H><02H><HEADER>
<00H><03H><LINE NUMBER><COLOR><RUN LENGTH><COLOR><RUN LENGTH>.....
<00H><03H><LINE NUMBER><COLOR><RUN LENGTH><COLOR><RUN LENGTH>.....
.....
.....
(repeat for a total of 256 lines)
<00H><06H><LEGEND NUMBER><LEGEND TEXT>
.....
.....
<00H><04H>
    
```

where :

- (a) <PICTURE TYPE> is a byte indicating the type of radar image :

Image	<PICTURE TYPE>	
TEST	1	; test pattern
CAPLO	2	; low resolution CAPPI
ECHOTO	3	; low resolution echo top map
ETPPI	4	; echo top map
ACCUMA	5	; accumulation map
VERTCX	6	; vortex track
PTCAST	7	; point forecast
CAP256	8	; 3 km CAPPI at 256 km range
CAP512	9	; 3 km CAPPI at 512 km range

- (b) <HEADER> is 40-byte text title of the radar image including time and date.
- (c) <LINE NUMBER> is a byte representing the image line number.
- (d) <COLOR> is a 1-byte colour code for the next <RUN LENGTH> pixel(s) defined as follows :

<COLOR>	
01H	coast line
02H	5 to 10 mm/hr
03H	0.3 to 0.5 mm/hr
04H	1 to 2 mm/hr
05H	100 to 200 mm/hr
06H	20 to 30 mm/hr
07H	ground
08H	background
09H	not used
0AH	10 to 20 mm/hr
0BH	0.5 to 1 mm/hr
0CH	2 to 5 mm/hr

0DH	50 to 100 mm/hr
0EH	30 to 50 mm/hr
0FH	>200 mm/hr
10H	land/sea

- (e) <RUN LENGTH> is a byte indicating 1 to 255 pixels of the same colour.
- (f) <LEGEND LINE> is a number from 01H to 16H indicating the row number of the text line.
- (g) <LEGEND TEXT> is a line of text with a maximum length of 32 bytes.

A.2 Guangzhou radar image format

There are four kinds of radar image files : PPI, RHI, LOCAPPI and TRACK. Each radar image file is made up of an image header and echo data.

The image header has 128 bytes and is divided into the following fields :

```
<type><level><unit><year><month><date><hour><minute>
<second><time><radius><EL><cappiHT><locenterx><locentery>
<disnce><vcsx><vcsy><trackx><tracky><tspeed><AZ><VRHT>
<temp>
```

where :

- (a) <type> is a byte indicating the type of the radar image.

Image	<type>	Kind
PPI	1	PPI ; PPI
RHI	2	RHI ; RHI
CAPPI	3	PPI ; CAPPI
ETPPI	4	PPI ; echo top PPI
TCW	5	PPI ; total column water content
ZPPI	6	PPI ; elevated PPI
PRE	7	PPI ; precipitation
LOCAPP	8	LOCAPP ; sectored CAPPI
VCS	9	RHI ; vertical cross-section
TRACK	10	TRACK ; vortex track

- (b) <level> has 16 bytes indicating the 16 colour reference values.
- (c) <unit> is a byte indicating the unit of <level>.

<unit> Unit

1	dbz
2	km
3	mm
4	cm
5	0.1 g m ⁻³

- (d) <year> has 4 bytes with the first byte indicating the last two digits of the year. The remaining 3 bytes are used for indicating the years of 3 other tracking points in the case of a TRACK type image.

- (e) <month> has 4 bytes with the first byte indicating the month. The remaining 3 bytes are used for indicating the months of 3 other tracking points in the case of a TRACK type image.
- (f) <date> has 4 bytes with the first byte indicating the day. The remaining 3 bytes are used for indicating the dates of 3 other tracking points in the case of a TRACK type image.
- (g) <hour> has 4 bytes with the first byte indicating the hour. The remaining 3 bytes are used for indicating the hours of 3 other tracking points in the case of a TRACK type image.
- (h) <minute> has 4 bytes with the first byte indicating the minute. The remaining 3 bytes are used for indicating the minutes of 3 other tracking points in the case of a TRACK type image.
- (i) <second> has 4 bytes with the first byte indicating the second. The remaining 3 bytes are used for indicating the seconds of 3 other tracking points in the case of a TRACK type image.
- (j) <time> is a byte between 0 and 60 indicating the rainfall accumulation period in minutes.
- (k) <radius> has 1 word (i.e. 2 bytes) indicating the radius of PPI display in km.
 <radius> = 75, 150, 300, or 512
- (l) <EL> is a word indicating the elevation of PPI display in 0.1.
- (m) <cappiHT> has 4 words with the first word indicating the height of the CAPPI in m. For LOCAPPI, the remaining 3 words indicate the heights of the other 3 LOCAPPI sectors.
- (n) <locenterx> is a word indicating the centre x-coordinate of LOCAPPI in km.
- (o) <locentery> is a word indicating the centre y-coordinate of LOCAPPI in km.
- (p) <distance> is a word indicating the horizontal distance of a LOCAPPI or VCS type image.
- (q) <vcsx> has 2 word with the first word indicating the start x-coordinate of a VCS image in km. The last word indicates the end x-coordinate of the VCS image in km.
- (r) <vc sy> has 2 words with the first word indicating the start y-coordinate of a VCS image in km. The last word indicates the end y-coordinate in km.
- (s) <trackx> has 4 words indicating in order the first, second, third and fourth echo tracking x-coordinates in km.
- (t) <tracky> has 4 words indicating the first, second, third and fourth echo tracking y-coordinates in km.
- (u) <tspeed> has 4 words with the first three indicating the calculated speed in km/hr between successive tracking points.
- (v) <AZ> is a word indicating the azimuth of a VCS image in 0.1.

- (w) <VRHT> is a word indicating the height of a RHI or VCS image in km.
- (x) <temp> is not defined.

The radar image data frame has one of the following four formats depending on the kind of image :

- (a) PPI type image

The image is made up of 458 pixels x 512 lines x 16 colours. Each line of the image has the following sequence of data :

<countN><color1><color2>...<colorN><length1><length2>...<lengthN>

where <countN> is a word indicating the total number of colour segments in this line. <color1>, <color2>, ..., <colourN> are 1 byte each indicating the colour value of each segment. <length1>, <length2>, ..., <lengthN> are 1 byte each indicating the corresponding length of each segment in pixels.

- (b) RHI type image

The image is made up of 360 pixels x 240 lines x 16 colours. Each line of the image has the same data format as that of the PPI type image.

- (c) LOCAPPI type image

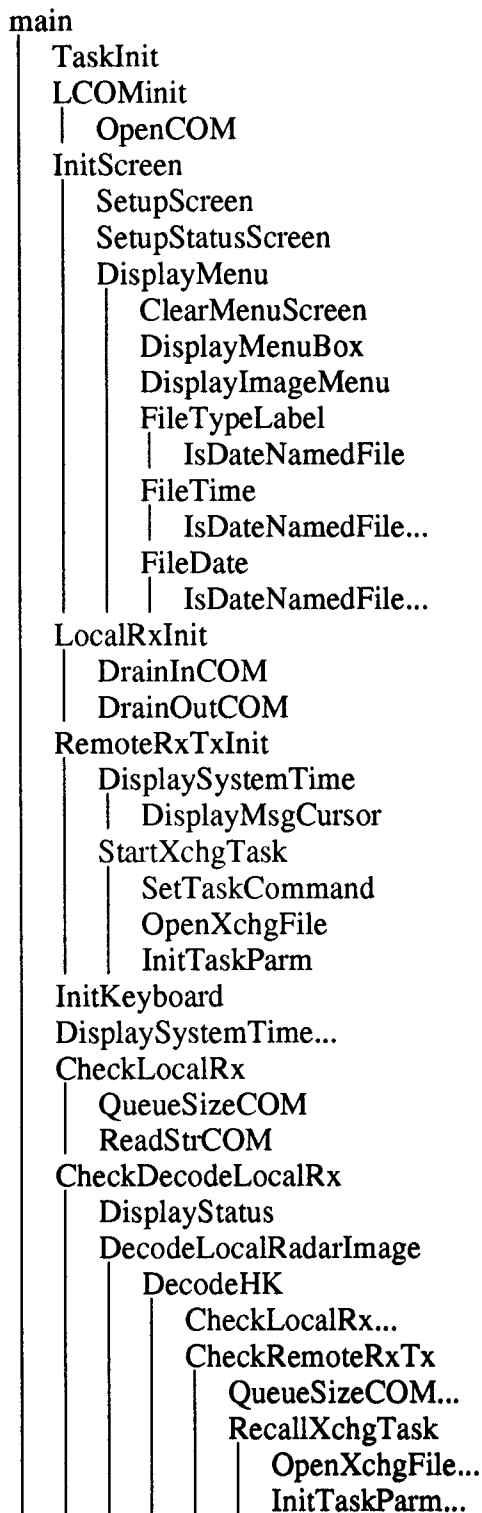
The image is a made up of 4 sub-images each of 160 pixels x 160 lines x 16 colours. Each line of a sub-image is coded in the same way as the PPI type image.

- (d) TRACK type image

There are no image data. Track parameters are given in the image header.

APPENDIX B - Program call tree

The following diagram shows the program call tree for the radar image exchange program. Starting from first column, the program module 'main' invokes initialization subroutines and then enters a task loop shown in the second column. The third column onward show the functions or procedures being called within these tasks. The symbol '...' shows that the function or procedure is re-used.



```

KillCurrentRemoteTask
|
| DrainOutCOM...
| DrainInCOM...
| StopXchgTask
| | DrainOutCOM...
| | CloseXchgFile
| OutputCOM
| WriteStrCOM
ReadStrCOM...
ConnectString
OutputCOM...
OpenXchgFile...
InitTaskParm...
EndXchgTask
| CloseXchgFile...
| FileTime...
| DisplayStatus...
| IsDateNamedFile...
| CloseCOM
| LCOMinit...
| SetupStatusScreen...
| DisplayMenu...
CheckCommandRetry
| OutputCOM...
| StopXchgTask...
WriteChrCOM
StopXchgTask...
CheckWAITDATATXACKretry
| StopXchgTask...
| OutputCOM...
DrainOutCOM...
CheckLastCode
ChkBlkNum
WriteStrCOM...
Checksum
DrainInCOM...
closegraph
pixel
DecodeGZ
DecodeGZradar
| initppi
| | drawlandmark
convtdata
disptype
printA
ppiparam
| convtdata...
ppidisplay
| convtdata...
| closegraph...
CheckLocalRx...

```

				CheckRemoteRxTx...
				pixel...
				trackparam
				convtdata...
				initppi...
				crossword
				initlocappi
				locparam
				convtdata...
				locappidisplay
				CheckLocalRx...
				CheckRemoteRxTx...
				convtdata...
				closegraph...
				pixel...
				initvcs
				rhiparam
				convtdata...
				vcdisplay
				CheckLocalRx...
				CheckRemoteRxTx...
				convtdata...
				closegraph...
				pixel...
				vcsparam
				convtdata...
				MakeFileName
				IsDateNamedFile...
				CheckOutputSchedule
				StartXchgTask...
				SetXchgTaskInQ
				strcpy?
				SetTaskCommand...
				CheckRemoteRxTx...
				CheckDecodeRemoteRx
				DisplayStatus...
				DecodeRemoteRadarImage
				DecodeHK...
				DecodeGZ...
				MakeFileName...
				IsDateNamedFile...
				CheckKeyboard
				CheckMenu0
				kbchar
				DisplayMenu...
				DecodeHK...
				DecodeGZ...
				SetupStatusScreen...
				DisplayStatus...
				StartXchgTask...
				CtrlChandler

```

| | CloseCOM...
CheckMenu1
| DecodeHK...
| DecodeGZ...
| DisplayMenu...
| ClearMenuScreen...
| DisplayMenuBox...
| PopMenu
| | DisplayMenu...
CheckMenu2
| GetKey...
| DecodeHK...
| DecodeGZ...
| DisplayMenu...
| ClearMenuScreen...
| DisplayMenuBox...
| PopMenu...
CheckMenu3
| GetKey...
| DisplayMenu...
| ViewNextPage
| | EraseToLineEnd
| | EraseToScreenEnd
| | EraseToLineEnd...
| PopMenu...
EditFileSpec
| GetKey...
| PopMenu...
| DisplayMenuBox...
| DisplayMenu...
| StartXchgTask...
| ViewNextPage...
| DecodeHK...
| DecodeGZ...
| ClearMenuScreen...
| MakeFileName...
| IsDateNamedFile...
CheckConfirm
| kbchar...
ClearMenuScreen...
DisplayMenuBox...
PopMenu...
KillCurrentRemoteTask...
KillDelayedRemoteTask
KillQueuedRemoteTask
CheckViewPage
| GetKey...
| ClearViewWindow
| | EraseToLineEnd...
| PopMenu...
| ViewFirstPage

```

```

| ViewNextPage...
ViewLastPage
| ViewPrevPage
| | ViewNextPage...
ViewPrevPage...
ViewNextPage...
ViewPrevLinePage
| ViewNextPage...
ViewNextLinePage
| ViewNextPage...
CheckMenu4
| GetKey...
ClearViewWindow...
DisplayMenu...
DisplayMessage
| EraseToLineEnd...
| EraseToScreenEnd...
ViewLastPage...
PopupMenu...
CheckEditMsg
| GetKey...
ClearViewWindow...
PopupMenu...
DisplayMenu...
DisplayMessage...
StartXchgTask...
ClearViewWindow...
DisplayMenu...
DisplayStatus...
CheckOutputSchedule...
HKOutputTimeSlot
ClearEditWindow
ClearMsgWindow
Locator
| pixel...

```