GBPPR 'Zine

Issue #116 / The Monthly Journal of the American Hacker / December 2013

"Find a career that you are interested in, either by searching by name or keyword or based on a school subject that you enjoy. When you find one that you are interested in, click SAVE TO MY PLAN. You are not allowed to select Professional Athlete as one of your saved careers. If you are interested in a career in athletics, research a career that does not require you to be a pro–athelete."

––– Excerpt from a local public school homework assignment for 6th graders researching a career. Are people starting to wake up?

Table of Contents

BELL SYSTEM PRACTICES
AT&TCo SPCS

SECTION 231-045-415
Issue 1, August 1979

10A REMOTE SWITCHING SYSTEM—CALL PROCESSING SOFTWARE SUBSYSTEM DESCRIPTION 2-WIRE NO. 1 AND NO. 1A ELECTRONIC SWITCHING SYSTEMS

CONTENTS PAGE

Figures

CONTENTS PAGE

Tables

1. GENERAL

INTRODUCTION

1.01 The 10A Remote Switching System (RSS) call processing software performs the control functions necessary to operate a 10A RSS under control of a host No. 1 or No. 1A Electronic Switching System (ESS).

1.02 When this section is reissued, the reason for reissue will be given in this paragraph.

1.03 Part 5 of this section provides a defined list of the abbreviations and acronyms as used herein.

PURPOSE OF THE RSS SOFTWARE

1.04 The RSS call processing software provides an ESS central office the capability to supply ESS features to lines served by a ***remote*** switching network. This remote network, referred to as the remote terminal (RT), may be used as a central office replacement vehicle for small community dial offices or as a pair gain system to reduce the number of subscription pairs in the loop plant. Since most of the call processing functions for RSS lines are performed by the host ESS office, a full family of ESS features can be provided to lines which otherwise would have been limited to basic service.

SCOPE OF SECTION

1.05 This section provides an introduction to the RSS software operating in a No. 1 or No. 1A ESS. Information unique to a specific system application is so noted.

1.06 This section is based on the 1E6 (No. 1 ESS) and 1AE6 (No. 1A ESS) versions of the generic program.

2. PIDENTS DESCRIBED IN SECTION

2.01 Table A provides a PIDENT to program number cross-reference for the RSS PIDENTs described in this document. Table A is not an exhaustive list of all RSS PIDENTs, rather only those PIDENTs which form the call processing core of the RSS software are listed.

2.02 A brief introduction to each RSS PIDENT is given below. Detailed information is provided in subsequent parts of this document and in the program listings.

(a) The remote order buffer (ROB) failure program, PIDENT ROBF, processes failure actions and reports having to do with failure conditions associated with ROB processing.

(b) The ROB loading and administration program, PIDENT RMSG, provides loading, activating, and other client services on request.

(c) The ROB execution program, PIDENT ROBE, administers the interface between the ROBs and data links to the RSS.

(d) The RSS message routing program, PIDENT RRTE, is responsible for unloading messages from the data link receive buffers in the call store (CS).

(e) The RSS network administration program, PIDENT RNAD, performs the actions necessary to control path selection and related control activities associated with the remote network.

(f) The RSS terminal administration program, PIDENT RTAD, consists of a collection of routines which controls channel administration.

(g) The RSS disconnect program, PIDENT RDIS, handles disconnect reports and actions required for disconnecting the RT portion of a call.

(h) The RSS supervision report program, PIDENT RSUP, unloads and processes RSS supervision hopper entries. Hopper entries are of two types: one type is for a change in supervision on an RSS line terminal; the other type is a miscellaneous scanner number report.

(i) The RSS reswitch program, PIDENT RESW, controls the ***reswitch-up*** or ***reswitch-down*** actions associated with intra-RSS calls.

(j) The RSS dialing connection program, PIDENT DCNR, is responsible for the actions required to process an RSS origination, including path selection and activation.

(k) The RSS unit translation update program, PIDENT RUTU, controls the RT translation update task. Translation information is transferred from the ESS to the RT via data link messages.

(l) The RSS traffic program, PIDENT RTRF, provides a means of collecting status information necessary for usage or traffic counts against the RSS.

TABLE A

RSS PIDENTS

PIDENT	TITLE	NO. 1 PR-	NO. 1A PR-
ROBF	ROB Failure	1A499	6A499
RMSG	ROB Loading and Administration	1A600	6A600
ROBE	ROB Execution	1A601	6A601
RRTE	RRS Message Routing	1A602	6A602
RNAD	RSS Network Administration	1A610	6A610
RTAD	RSS Terminal Administration	1A611	6A611
RDIS	RSS Disconnect	1A612	6A612
RSUP	RSS Supervision Report	1A613	6A613
RESW	RSS Reswitch	1A614	6A614
DCNR	RSS Dialing Connection	1A130	6A130
RUTU	RSS Unit Translation Update	1A633	6A633
RTRF	RSS Traffic	1A634	6A634

3. FUNCTIONAL DESCRIPTION

GENERAL

3.01 The RSS software provides a means of controlling a remotely located switching network. This system takes advantage of existing equipment and control capacity in an ESS to provide a more cost effective switching entity for central offices than provided by conventional stand-alone switches. While the bulk of the software controlling the RSS resides in the ***host*** ESS, the smaller and less sophisticated microprocessor with its associated firmware located in the remote terminal functions as an autonomous peripheral. A major advantage of this arrangement is that the complex tasks of call processing, using existing software facilities and features, are performed in the host ESS. RSS also furnishes a means of providing ESS services to customers served by electromechanical offices. As shown in Fig. 1, RSS consists of a host ESS office, a remote terminal, and interconnecting voice and data links. Note that a particular ESS may ***host*** one or more RSSs; the data and voice links may be other than metallic paths.

3.02 The data links are used for communication between the ESS central control (CC) via the peripheral unit controller and the microprocessor in the RT. Information such as line origination, acknowledgment, and status (RT to ESS), and network orders (ESS to RT) is passed via the data link.

BASIC PATH CONNECTIONS

3.03 When an RSS line goes off-hook, an origination message is given to the ESS via the data link. A connection is established from the remote equipment number (REN) to the customer dial pulse receiver (CDPR) as shown in Fig. 2A. For the case where the called party is served by the host ESS, a ringing connection is established as shown in Fig. 2B. Call handling progresses as for a normal ESS line-to-line (L-L) connection. Answer detection results in a talking path as shown in Fig. 3B.

3.04 For calls terminating to an RSS line, the ringing connection for both parties is shown in Fig. 3A. Ringing voltage is supplied to the called RSS line via the universal service circuit (USC); answer supervision is done by the ESS ring circuit. Normal ESS ringing, with ringing voltage supplied by the ESS ringing circuit, is not permitted for RSS terminations since the ringing path is through the solid state RT network. The ringing circuit is used for answer supervision since this allows ring trip to be detected in the usual fashion. After ring trip, the talking path is connected as shown in Fig. 3B.

3.05 For an intra-RSS call, the ringing connection is established as shown in Fig. 4A. Note that a talking path is first reserved and is then connected after ring trip. At this time, for a simple two-party intra-RSS call, the ESS resident RSS call processing programs calculate and reserve a unique RSS path to replace the existing path through the ESS and RSS (Fig. 4B). A ***reswitch-down*** operation results in the intra-RSS talking path shown in Fig. 4C. This reswitch operation frees the paths and equipment used in the initial talking connection.

3.06 Once in the reswitch-down configuration, supervision is done at the RT. Flash from a line with flash privileges results in a reswitch-up, allowing the ESS access to the line requesting special service. In this manner, ESS facilities are used only as necessary.

PROGRAM ARCHITECTURE

3.07 Figure 5 provides an overview of the major partitioning of the RSS software and shows some of the interfaces with existing ESS structures. The RSS PIDENTs have been related to particular functions where possible. The interconnecting lines illustrate functional relationships and may indicate program control flow.

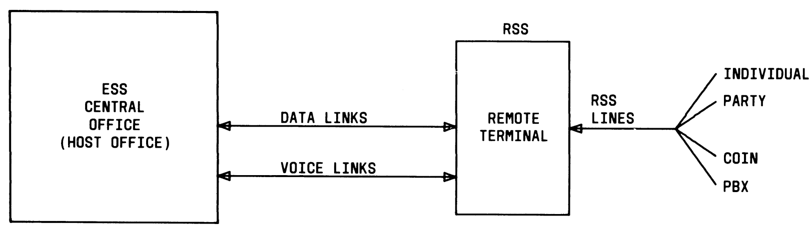
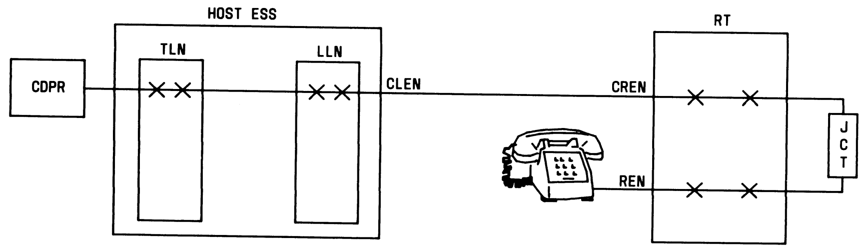
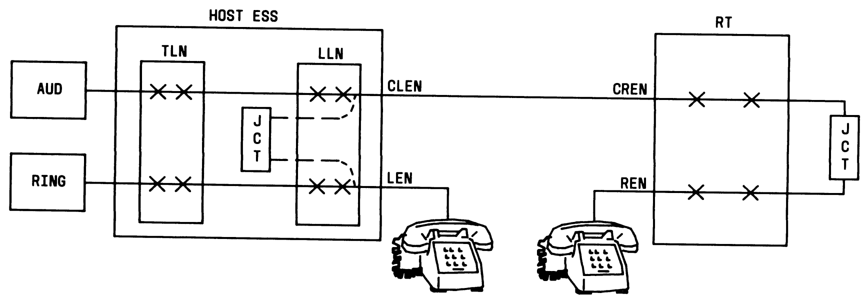


Fig. 1—RSS

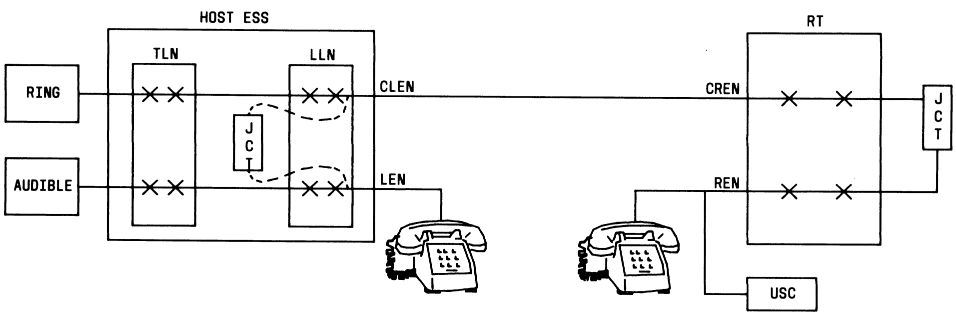


A. DIGIT COLLECTION

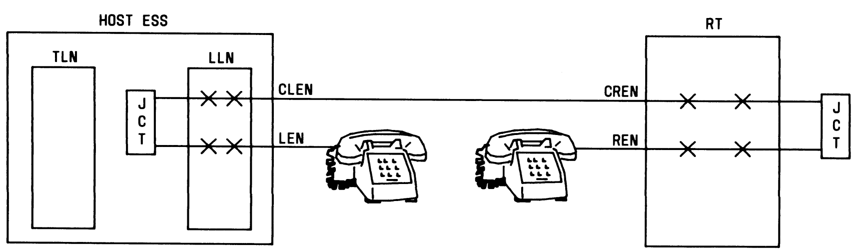


B. RINGING PATH, RESERVED TALKING PATH

Fig. 2—Call Originating from RSS

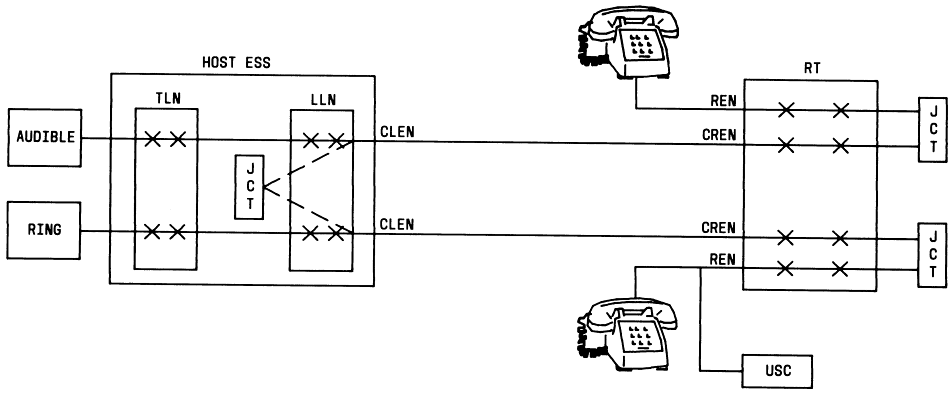


A. RINGING CONNECTION, RESERVED TALKING PATH

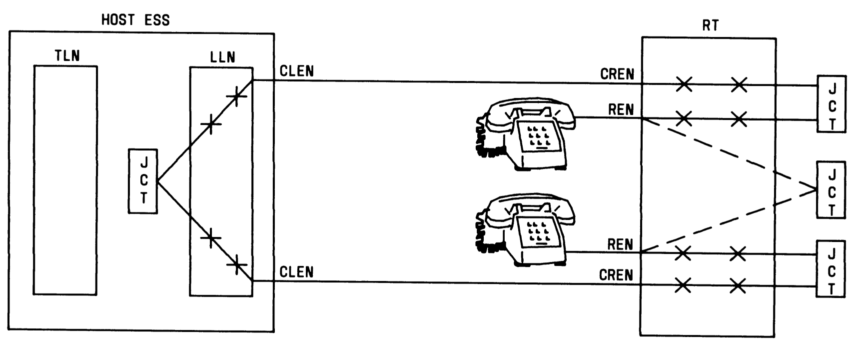


B. TALKING PATH

Fig. 3—Call Terminating to RSS



A. RINGING CONNECTION, RESERVED INITIAL TALKING PATH



B. INITIAL TALKING CONNECTION

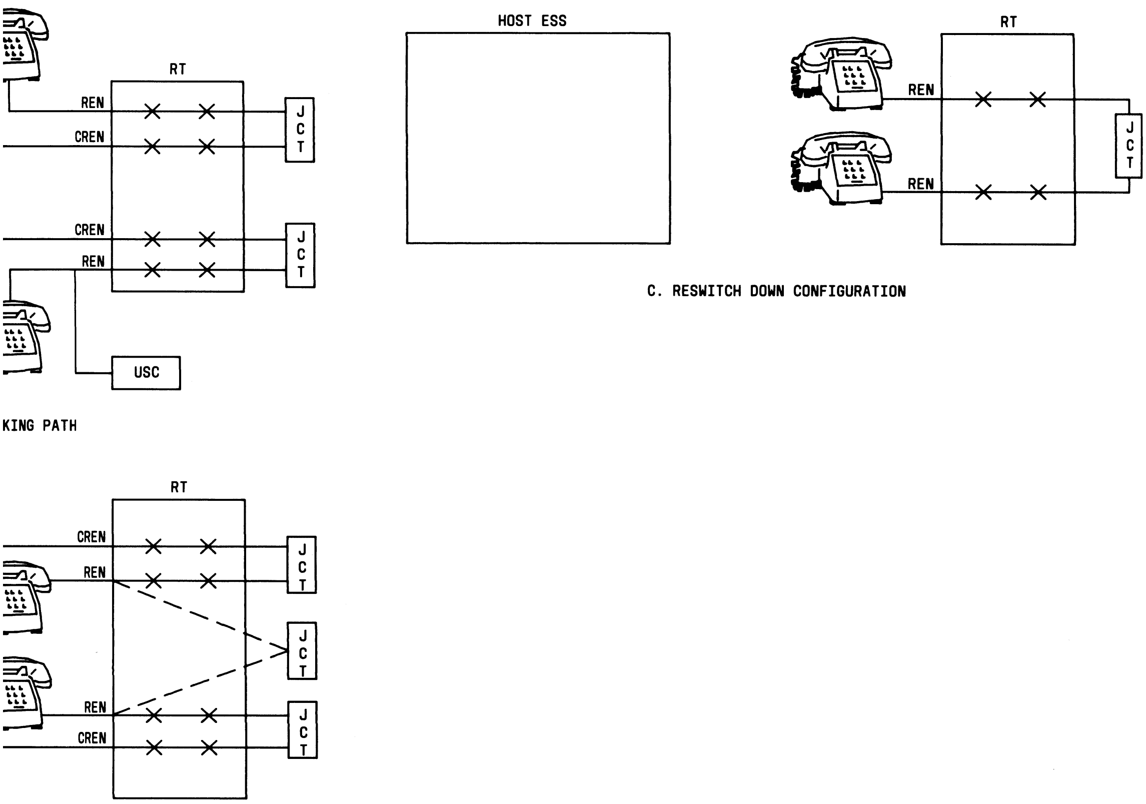
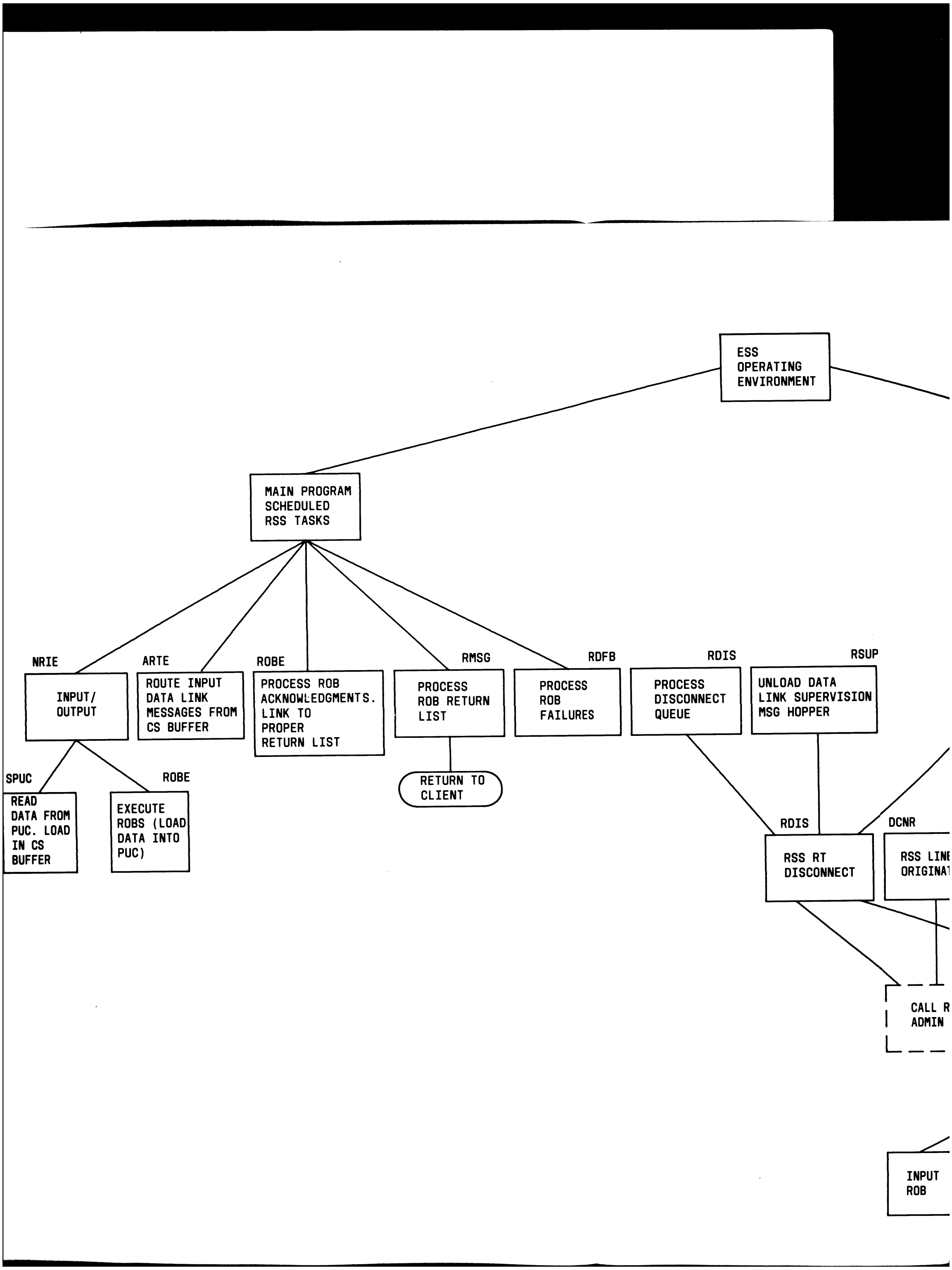


Fig. 4—Intra-RSS Call


ESS OPERATING ENVIRONMENT
MAIN PROGRAM SCHEDULED RSS TASKS
NRIE
INPUT/ OUTPUT
ARTE
ROUTE INPUT DATA LINK MESSAGES FROM CS BUFFER
ROBE
PROCESS ROB ACKNOWLEDGMENTS. LINK TO PROPER RETURN LIST
RMSG
PROCESS ROB RETURN LIST
RDFB
PROCESS ROB FAILURES
RDIS
PROCESS DISCONNECT QUEUE
RSUP
UNLOAD DATA LINK SUPERVISION MSG HOPPER
SPUC
READ DATA FROM PUC. LOAD IN CS BUFFER
ROBE
EXECUTE ROBS (LOAD DATA INTO PUC)
RETURN TO CLIENT
RDIS
RSS RT DISCONNECT
DCNR
RSS LINE ORIGINAT
CALL R ADMIN
INPUT ROB

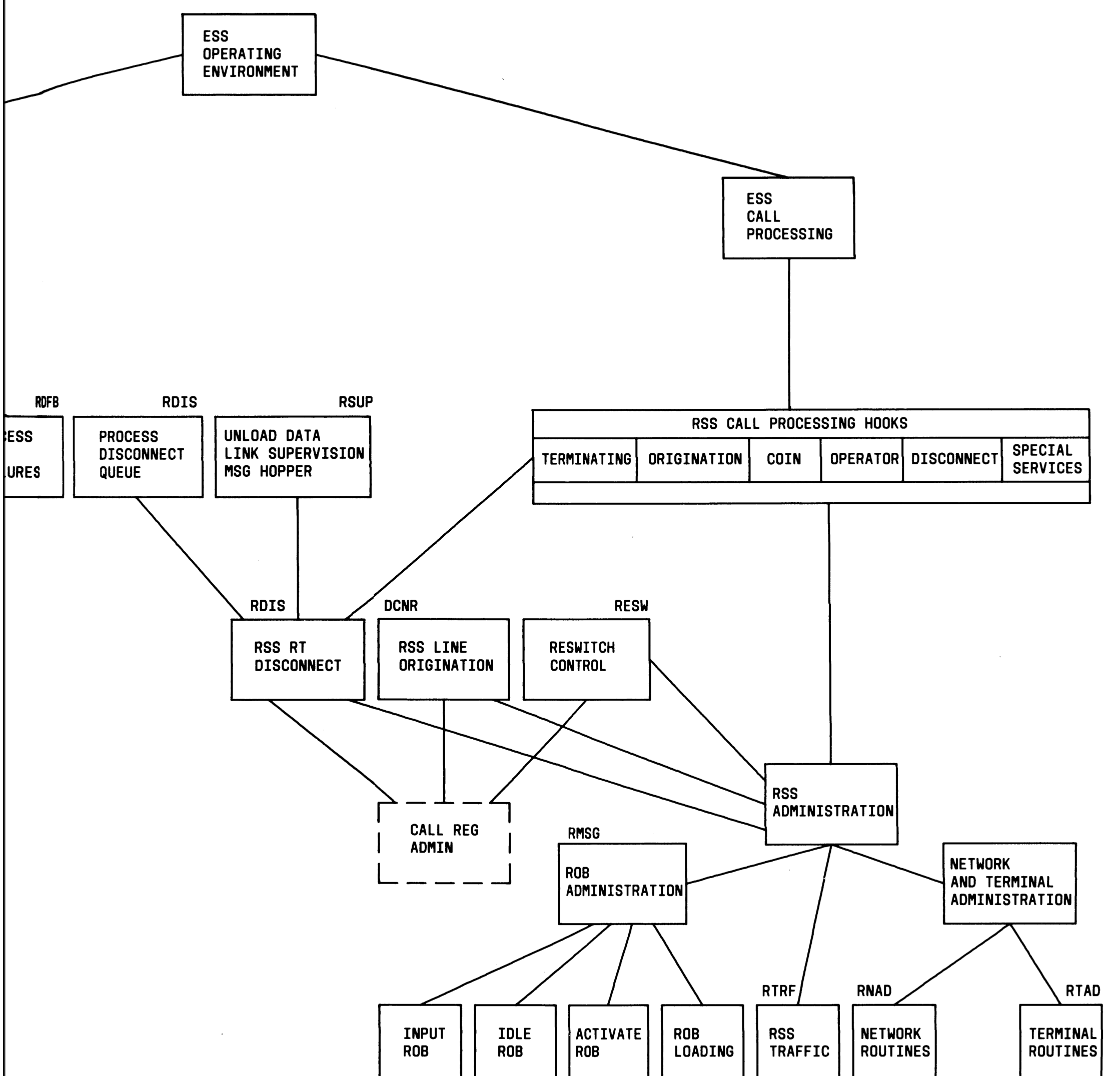


Fig. 5—RSS Call Processing Structure

Input/Output

3.08 A major portion of the input/output (I/O) work is concerned with the sending and receiving of data link messages. For input, data is read from the peripheral unit controller (PUC) into the ESS CS receive data buffers. In No. 1 signal processor (SP) offices, the receive data buffers (one per RSS) reside in signal processor CS. RRTE is the only signal processor program in the RSS feature. The ROB execution module is responsible for executing orders previously loaded into an ROB. PIDENT ROBE transfers message data from the ROBs to the PUC transmit buffers and executes ROB control subroutines.

Scheduled Non-I/O Tasks

3.09 ***Input Routing:*** Input routing of data link messages is concerned with processing the incoming data stored in the CS receive buffers. This data is delimited into messages, type determined (ie, supervisory, acknowledgment, TTY, audit, etc), and then stored into the corresponding client buffer.

3.10 ***Processing ROB Returns:*** ROB return list processing involves, first, placing completed ROBs on a return list and giving control to the client success or failure address as required. For ROBs linked to success lists, control is returned to the client program at the success address stored in the ROB. The call register address (CRA) is also given to the client at this time. For ROBs linked on the maintenance unexpected return list (MURL) and if specified by failure option, the call will be cleared from the system and control returned to the client's failure address. ROBs on the call unexpected return list (CURL) result in control being given to the unexpected result transfer address (URTA) stored in the ROB.

3.11 ***Unload RSS Supervision Hopper:*** The function here is to process supervisory data link messages from the RSS supervision hopper. The control routines first determine the message type (ie, line supervision or scan point report) and for supervision message use the status of the remote equipment number (REN) to determine which program should process the report and then pass control to that program. If an RSS call register is on the call, the report is made to the program in control of the call register. If there is no call register and the REN is busy, control is given to the RSS disconnect program for on-hooks or to the reswitch program for flashes. RSS scan point reports are handled similar to that in the host.

3.12 ***Process Disconnect:*** RSS disconnect actions require operations both common to the ESS and unique to the RSS. The disconnect program processes the disconnect queue that is set up as a result of requests from the ESS disconnect routines that want to hand off the disconnect actions for the RT portion of the call. If the facilities to handle the disconnect are not available at the time of the request, the request is queued. The disconnect routines also handle the condition where an on-hook data link message is received for an intra-RSS call.

3.13 ***Origination:*** The RSS origination module obtains control of the call after the off-hook (origination) data link message is received for the idle REN and entered in the host line service request hopper. When the host line service request hopper is unloaded, RENs are handed off to the RSS dialing connection module.

3.14 ***Reswitch:*** Reswitching refers to either reconfiguring an intra-RSS call that is switched through the ESS (Fig. 4B) to a simple intra-RSS path (Fig. 4C) ***or*** to the reverse of this process. A reswitch-down is requested by the ESS call processing programs after the initial talking path has been established. A reswitch-up may be requested from the following sources:

(a) The unload data link supervisory message routine if a flash is received on an intra-RSS call.

(b) Existing ESS programs supplying services such as call waiting, operator no-test, etc, may find it necessary to request a reswitch-up for an intra-RSS call since these services are available only via a host connection.

3.15 ***ROB Administration:*** The ROB Administration programs contain a number of routines enabling user programs to hunt, idle, activate, and load ROBs with the various orders available for transmission to the RSS and provides administration data necessary to process these.

3.16 ***RSS Network and Terminal Administration Programs:*** This collection of routines, user-called as necessary, performs remote network and terminal management functions. Network management routines are provided to reserve, set up, tear down, and erase RSS network paths. Provision is also made for path memory administration. Other routines are available to provide client program access to the remote network path memory data base. Terminal administration also includes the user interface routines to modify or retrieve the status of a line, channel, or tone circuits. These routines are used to hunt and idle terminals as well as to administer the terminal records for maintenance purposes.

4. PIDENT FUNCTIONAL DESCRIPTIONS

RSS DIALING CONNECTION

PIDENT DCNR

4.01 The RSS dialing connection problem consists of the following steps:

(a) Hunt an idle channel to the host ESS.

(b) Hunt an idle receiver in the host.

(c) At the RT, connect a REN-to-channel REN (CREN) path.

(d) Operate sleeve leads as required.

(e) Perform party test if two-party.

(f) Request fast transfer of supervision from REN to CREN.

(g) Connect the receiver to the host end of the path.

(h) Activate digit collection.

4.02 The processing of an RSS origination is initiated by the RSS dialing connection program, PIDENT DCNR. DCNR unloads the RSS service request hopper and puts the RENs into the host service request hopper. PIDENT DCNL gives control to DCNR for RENs found in the host service request hopper. An attempt is then made to seize an originating register (OR), a peripheral order buffer (POB) and a ROB (ie, a POB-ROB), and then begin to build the POB-ROB orders needed to connect the digit collection path shown in Fig. 2A. Queueing may occur for software resources such as ROBs or POBs; retry may occur for ESS network blockage; RT path blockage will dump the call. Digit collection is accomplished in the host ESS by normal digit collection routines. Prior to this, DCNR establishes a fast transfer of supervision state in the RT to ensure that no digits are missed. Once the digit receiver is connected to the line, DCNR passes control to PIDENT DCNL and digit collection proceeds as for a normal ESS line origination. Path selection for the REN to CREN connection is accomplished with a call to the RSS network administration program, PIDENT RNAD. If the ensuing path hunt is successful, DCNR asks PIDENT NCIN, the ESS network program, for a path from the digit receiver to the CLEN. If path selection fails at the RT end, the call is dumped. If path selection at the ESS end fails, a path rehunt is done at both ends. Path blockage for the second time in the ESS network will dump the call.

Nonfailure Conditions

4.03 It is possible that an origination message may be received for a REN that is already marked busy in memory. This can occur for the following reasons:

(a) A terminating call to the REN is being processed

(b) A previous origination for the REN is being processed

(c) The REN is on a receiver queue.

Regardless of the reason, no action is taken on the origination message. If the REN is busy in error, the audits will find it and clear it.

4.04 If the origination is from an unassigned REN, it will be treated as denied originating and placed in the ignore state. Host lines are left busy for an audit to find; however, if this were done without informing the RT and the REN was permanently off-hook, the limbo origination state would time out and keep sending origination messages.

4.05 Lines in the plug-up state are unplugged in the same way as for host lines.

4.06 An origination from a REN whose major class indicates a denied origination state is dumped; ie, the OR is released and the line is marked high and wet (via an entry to the permanent signal partial dial REN [PSPR] program).

4.07 While awaiting service in the hopper or on the POB-ROB queues, the REN may enter the abandon or dial before dial tone state. In the host, the line is scanned directly when unloaded from the hopper or queue. However, it is not possible to directly scan a REN without using a ROB. Using a ROB would only serve to increase the probability of these two conditions occurring. Instead, the first order in the ROB to set up the path will scan the line for 150 ms and if it is on-hook, no further execution is done. An URTA return is made and the call is dumped. These scans are done at a rate which will not turn dial pulses into an abandon; ie, in the presence of dial pulses at least one off-hook in 150 ms should be seen. A similar scan will be done in the POB before setting up the CLEN to receiver connection. By setting up the call in the presence of dial pulses the call can be taken to partial dial as opposed to the REN causing repeated originations.

Failure Conditions

4.08 Call failure for lack of an OR is an illegal condition under the 1+ rule. ORs are engineered one for each service circuit requiring one and one extra to use in deciding that receiver queuing is necessary. An OR should not be held busy without an associated service circuit. An audit routine will be called and no further action is taken. The RT will time-out and resend the origination message.

4.09 If the origination finds a condition of all POBs or ROBs busy, the OR is placed on the POB-ROB queue. Meanwhile, a reserved path to a CREN is hunted in the RT and a reserved path from the CLEN to a receiver is hunted in the ESS. Hunting the receiver will assure having one when the POB-ROB queue is served and does not break the 1+ rule by keeping an OR without its associated hardware. This method enhances the system's response speed since the POB-ROB queue is served in class A. It also tends to concentrate an origination overload at one point, the receiver.

4.10 If a path blocking condition is encountered in the RT, the call is dumped and a resend (of the origination message) will occur. This provides time for paths or channels to become idle; and the failure is counted.

Channel Origination

4.11 Host LENs with a channel major class are not allowed to originate. A channel origination results in the LEN being marked high and wet. Channel originations can occur since, on disconnect, the CREN is put on a disconnect queue and the CLEN's line bit is idled in the same real time segment. If the REN should reoriginate before the disconnect ROB is executed at the RT, off-hook may be repeated on the channel and detected at the host as an origination. This could occur if the data link or RT were overloaded.

Fast Supervision

4.12 Some calls will require that the REN be left in the fast transfer of supervision state for the duration of the call. A bit in the OR, 04RSF, is used to indicate that this supervision state be left on. For detailed information describing the dialing connection sequences, refer to the DCNR program listing (Table A).

INPUT MESSAGE HANDLING

PIDENT RRTE

4.13 Messages coming into the ESS from an RSS are temporarily stored in a data link (DL) receive buffer in CS. These messages are received from the PUC via the scanner answer bus. (For No. 1 SP applications, RRTE is executed by the SP; the DL receive buffer is in SP CS.) Since the ESS and RSS operate autonomously, the various input buffers must be unloaded at a fast enough rate to keep up with the RSS message transmission rate. As such, global MSGRTE in PIDENT RRTE receives a scheduled entry from the main program. PIDENT RRTE unloads messages from data link receive buffer in CS (Fig. 6) and routes the individual messages to the appropriate client buffers and, when necessary, flags the corresponding client program for execution at the appropriate main program level (A, B, C, D, or E). All messages are composed of 16-bit words and are delimited in the receive buffer by sync words which preceed the start of each message (Fig. 7). RRTE uses these sync words to establish message synchronization. Immediately following the sync word is the message header, identifying the destination client and the number of words in the message. After unloading the header specified number of words, RRTE should find either another sync word or the end of the buffer. Failing this, RRTE attempts to reestablish sync by finding the next sync word.

4.14 Received messages are assembled into the appropriate client program's hopper and the client will be notified when a complete message is present. All client hoppers have the standard load and unload head cells. A message will not be loaded into the client's hopper if the number of words indicated by the header are not present in the message or if the client buffer overflows on receipt of the message. Detailed information is provided in the RRTE program listing (Table A).

PIDENT RSUP

4.15 PIDENT RSUP unloads and processes entries in the RSS supervision hopper. The scheduled main program entry to RSUP is via global RSSUPV. On entry, RSUP checks the next hopper entry to be processed; each hopper entry is two words long. If the entry to be unloaded is found in the reset condition (ie, the first word of the entry is all zeros), the hopper is empty and control is given back to the main program. If the word is found to contain all ones, the pointer is at the bottom of the hopper. For this case the pointer is reset to the top of the hopper and the initial checks are repeated. An entry which is neither set nor reset is treated as a valid entry and will be processed. The entry is read from the hopper and stored; the hopper entry is zeroed; and the hopper pointer is incremented to the next slot. Note that each entry is ***two*** words long.

4.16 The RSS supervision hopper stores two types of entries:

(a) Supervision change on an RSS line terminal

(b) Remote miscellaneous scanner number report.

Hopper entries are structured so that the first word contains the RSS number. The second word contains the ***type*** of entry as follows:

(a) For a remote miscellaneous scanner number report, the second word contains the scan point number and type of report (ie, on- or off-hook).

(b) For a remote line supervision change, the second word contains the supervision REN number and the supervision type (ie, on-hook, off-hook, or flash).

For specific information concerning RSUP control operations, refer to the program listing (Table A).

ROB PROCESSING

PIDENT RMSG

4.17 PIDENT RMSG (Fig. 6) performs the ROB loading and administration functions for the RSS software. RMSG provides entries for the following POB/ROB administrative functions:

(a) Hunt ROBs

(b) Hunt ROB-POB combination

(c) ROB activation

(d) Activate POB-ROB combinations

(e) ROB loading

(f) ROB BGNBLK and PRCACK (Begin block and process acknowledgment)

(g) Processing ROBs and POB-ROB combinations placed on various return lists.

4.18 For return list processing, RMSG receives scheduled entries from the main program. These entries are:

(a) RHPSL—ROB high priority success list

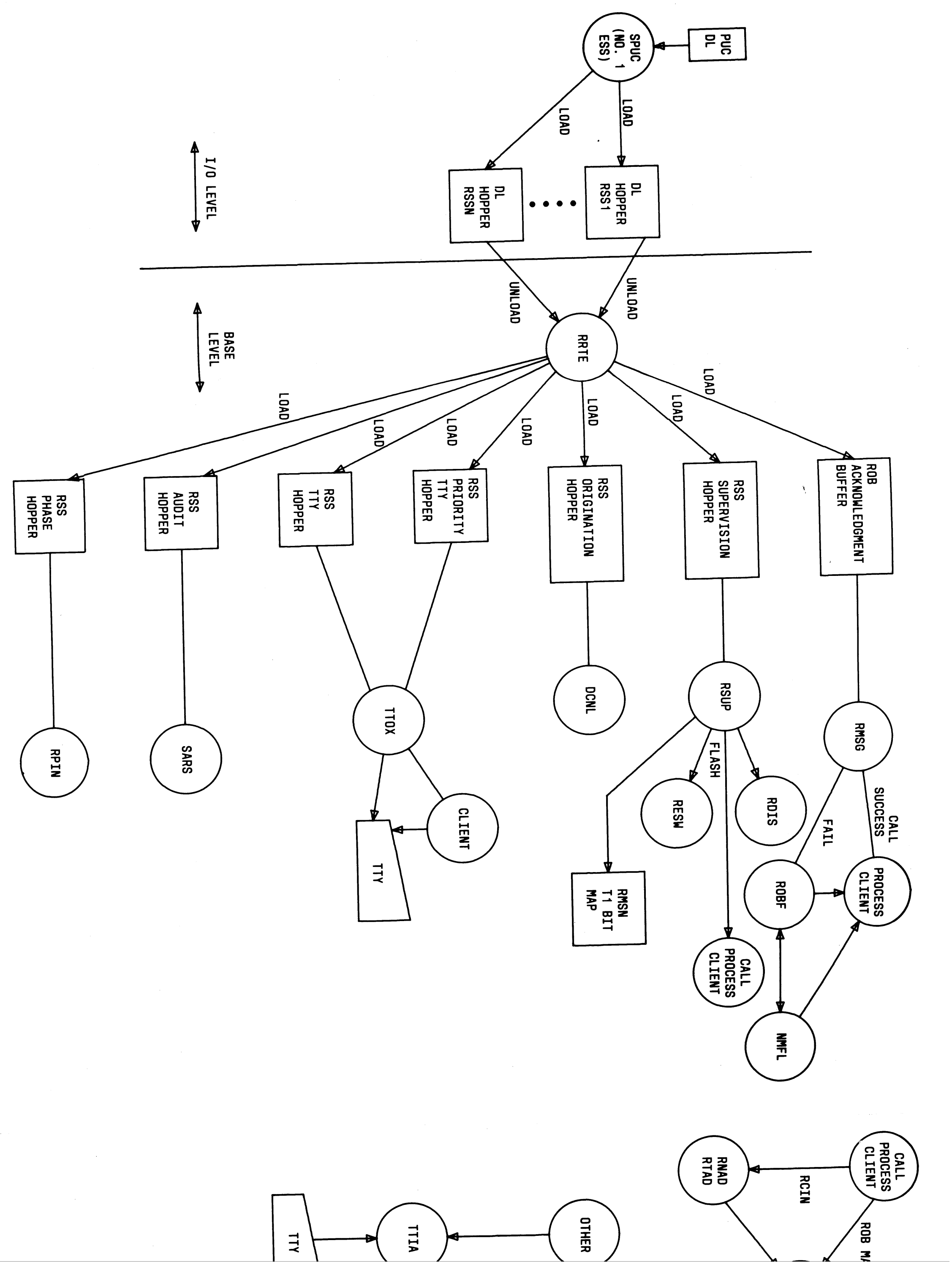
(b) PRHPSL—POB-ROB high priority success list

(c) RLPSL—ROB low priority success list

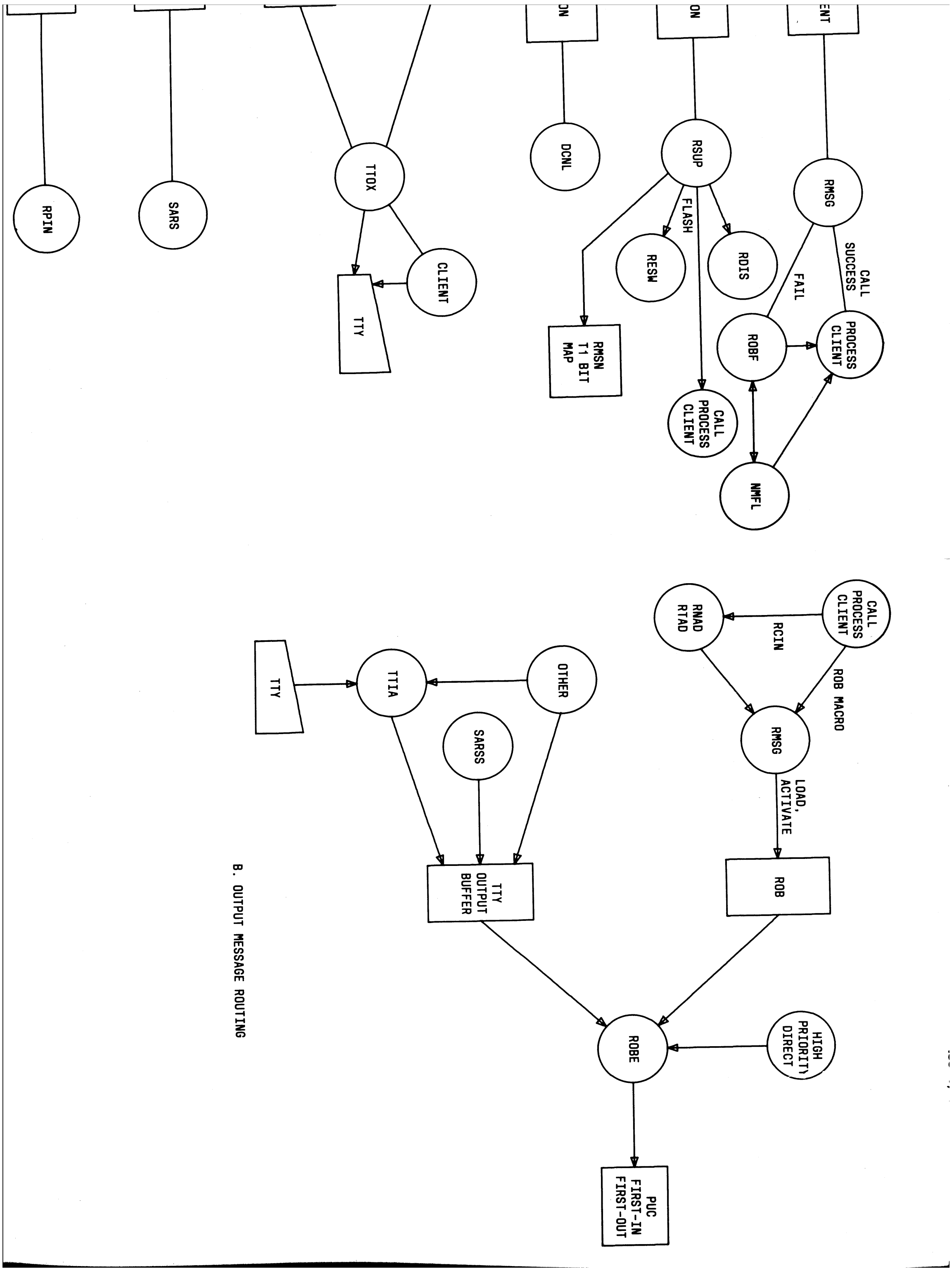
(d) PRLPSL—POB-ROB low priority success list

(e) ROCURL—ROB CURL.

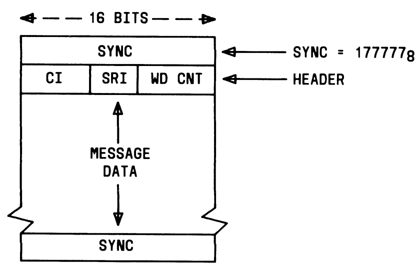
For detailed information describing the scheduled and client service routine entries, refer to the RMSG program listing (Table A).



A. INPUT MESSAGE ROUTING



B. OUTPUT MESSAGE ROUTING



HEADER:
CI = CLIENT IDENTITY = PRIMARY ROUTE TABLE INDEX
SRI = SUB ROUTE INDEX
WD CNT = NUMBER OF WORDS IN MESSAGE

Fig. 7—Message Format

PIDENT ROBE

4.19 ROBE is the ROB execution program for the RSS, providing the control routines necessary for PUC data link loading and administration. ROB execution is initiated via a scheduled class A entry from the ESS main program.

4.20 Prior to ROB execution, a main program entry to ROBE is made at global ROIW to determine the number of idle words in the PUC transmit data buffers for each RSS in the office. This idle word count is calculated from the number of words loaded (by the CC) versus the number of words unloaded (by the PUC). The idle word count (R1IDLCNT) is used later by the LODLNK routine. After idle word counts are determined for each RSS, ROBE returns control to the main program.

4.21 Also prior to ROB execution, ROBE receives a main program entry at global ROACKL. This routine is called to process ROBs on the acknowledgment timing list. ROBs are entered on this list while waiting for the RSS to send acknowledgment for receipt of a block of messages. If the acknowledgment arrives before an audit time-out occurs, the ROB is reactivated for ROBEXEC. If an audit time-out does occur, it is treated as a hardware failure and the ROB is put on the MURL.

4.22 After the acknowledgment timing list is processed, control falls into the ROB execution routine, ROBEXEC. This routine is responsible for processing entries in all active ROBs. Each entry to ROBEXEC results in all ROBs on the busy link list being processed, with inactive ROBs being passed over. The data entries are transferred to the PUC transmit data buffer and the control subroutines are executed. Individual ROBs are processed until inactivated by one of the control subroutines. When all of the ROBs have been processed, control is returned to the main program. For information describing these and other ROBE routines, refer to the program listing (Table A).

PIDENT ROBF

4.23 ROBF is the ROB failure action program for the RSS software. ROBF contains a set of control routines which manage the processing of ROBs which have experienced some type of execution failure. For instance, ROCFL and ROMURL are two scheduled entries from the main program. ROCFL is entered every nine seconds to time ROBs on the call failure timing list. RMURL is entered to remove ROBs from the MURL and to pass control to the appropriate failure routine as specified by the failure option in the ROB header. Global ROBFTD is entered from the POB failure program (NMFL). ROBF finds any RSS connections associated with the failed POB and restores them via blind-idle orders and cleans up the associated path memory. Control is then returned to NMFL to continue POB failure processing. For detailed information describing these and other ROBF routines, refer to the program listing (Table A).

RESWITCH FUNCTION

PIDENT RESW

4.24 Intra-RSS calls are initially completed to the talking state through the ESS network. An obvious savings in channels and associated host resources is realized if at some point during the call the connection can be reswitched down, out of the ESS network, to a simple RSS network path. Some infrequently occurring types of intra-RSS calls may not be reswitched.

4.25 Requests for a reswitch down are made by existing host ESS call processing programs at places when stable line-to-line calls are established and when both lines are possibly on the same RSS.

The reswitch-down function is attempted in the following cases:

(a) After ring trip

(b) After drop-back from a 3-port conference or attendant loop circuit.

The reswitch-down function is not attempted for the following:

(a) After release from an Operator No-Test connection

(b) If either RSS line is a coin line

(c) If an idle RSS call register cannot be found.

4.26 The reswitch-down process requires two steps. First is screening and initialization of possible reswitch candidates, and second is the action required to reswitch the call. The screening and initialization are performed by subroutine RSRDBA which does the following:

(a) Checks client's path memory for L-L and other valid connections.

(b) Checks that both input lines are CLENs to the same RSS.

(c) Neither line is a coin line.

(d) Seizes and initializes an RSS call register (but does not move path memory into the register).

(e) Links the RSS register to right of input register.

RSRDBA exits 0,J if reswitch-down has not been initialized or if reswitch-down cannot be done on the call. It exits on 1,J if reswitch initialization succeeds. RSRDBA is called after the L-L connection has been set up but before the change in network (CIN) to move path out of the client register.

4.27 The reswitch program obtains control after answer detection. The sequence of events is as follows:

(a) Reswitch-down is done with the scan mode of both RENs set to scan for disconnects.

(b) Answer timing is performed.

(c) The scan mode of any REN with flash privileges is changed to FLASH.

(d) The call is set to stable.

If the request to reswitch-down comes from a drop-back report as a result of moving the call off of a conference circuit, answer timing is not performed and the REN is set to the proper scan mode (FLASH OR DISCONNECT) during the reswitch process.

4.28 In order to miminize the noise and open interval during reswitch-down, the sequence is carried out as follows:

(a) All necessary resources (ROB, POB, paths) are seized.

(b) A ROB is loaded to disconnect the two line RENs (LRENs) from the CRENs, connect the two LRENs, and set up the proper supervision mode.

(c) A POB is loaded to disconnect the ESS network path and restore the CLENs.

(d) The ROB-POB is executed with the ROB executing first. There is a new failure option in the POB so that it does not tear down the RSS path if the POB fails.

In order to maximize utilization of the existing call processing functions provided in the host ESS, intra-RSS calls are reswitched up into the ESS to provide the more complex custom calling features. Some conditions requiring a reswitch up are:

(a) A REN flashes for add-on (three-way calling).

(b) Call waiting to a REN.

(c) Operator no-test, busy verify.

4.29 The reswitch-up process is carried out in a manner that minimizes the noise introduced into the talking connection and the open interval. The general procedure is as follows:

(a) Reserve all facilities (paths, channels)

(b) Set up the connection through the ESS network.

(c) Request the RSS to send an off-hook signal on both channels.

(d) Request RSS to switch the paths.

(e) Idle old facilities.

4.30 The major reswitching routines are:

(a) RESWITCH_UP

(b) RESWITCH_DOWN.

For detailed information describing the reswitch and supporting routines, refer to the program listing (Table A).

NETWORK AND TERMINAL ADMINISTRATION

PIDENT RNAD

4.31 The RSS network administration program, PIDENT RNAD, provides a collection of remote change in network (RCIN) routines for path administration in the RSS network. The RCIN routines provided by RNAD may be thought of as the functional equivalent of the ESS network CINs. Administration for the RSS network map is provided by the host ESS.

4.32 RCINs are called as subroutines for any call processing client requesting a path or path change in the remote network. The RCINs provide for the following operations:

(a) RESERVE—reserves an idle path in memory.

(b) CONNECT—reserves an idle path and builds the ROB orders necessary for network operation.

(c) ERASE—idles a path in memory.

(d) TEARDOWN—loads a ROB with orders to release a path in the remote network.

(e) SET-UP—builds the ROB orders necessary for network operation.

(f) KEEP—keeps path in memory and hardware (ie, no change).

These operations may be thought of as state changes in both the network hardware and the stored memory image of the network. Figure 8 shows the state transitions among the possible states.

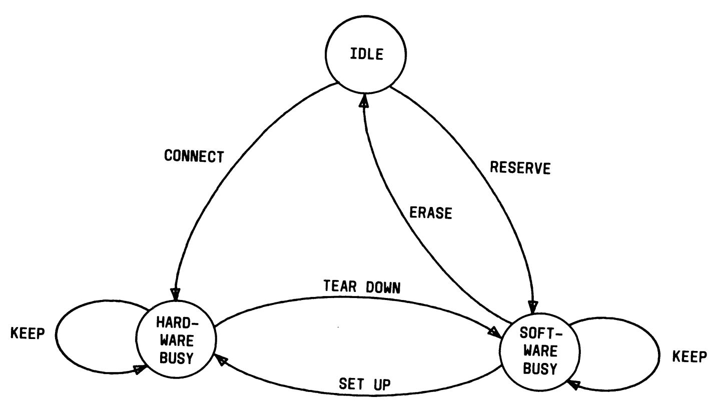


Fig. 8—RCIN Transitions

4.33 The success return from an RCIN means that the network map and path memory (in CS) have been updated to reflect a new RSS network state and that the necessary network orders (if any) to achieve this new state have been prepared and loaded into an ROB for execution. RCIN failure implies a blocked or busy condition in memory; the original path (if any) remains unchanged and no network orders are formed.

4.34 It is important to remember that RCIN success implies nothing about the success or failure of the network orders generated. It is possible that the ROB may encounter some difficulty and fail. Thus, even though an RCIN may be successful, the requested network actions could still fail. Some RCINs, specifically those which do not require circuit or path hunts, are always successful and have no means of reporting failure. For those RCINs which can fail, a failure return to the client indicates the reason for failure. For specific RCIN information, refer to the program listing (Table A).

PIDENT RTAD

4.35 PIDENT RTAD, the RSS terminal administration program, provides a collection of circuit administration routines for RSS resources. Status and administration of channels is the responsibility of the ESS. Circuit administration includes finding idle circuits and channels, path memory and busy-idle bookkeeping, and busy circuit idling. For detailed information describing the circuit administration routines, refer to the RTAD program listing (Table A).

DISCONNECT HANDLING

PIDENT RDIS

4.36 Disconnect actions for RSS calls depend on the particular path involved. The simplest case is for the intra-RSS call, where the connection is totally within one RSS network and no channels or ESS network paths are involved. For this type of call, the RENs are supervised in the RSS. Hits, flashes, and disconnects are detected by the RSS; flashes and disconnects are reported to the ESS via data link messages. If ignored by the host, disconnect messages continue to be sent at a slow rate; flash is sent only once for each flash detection. Global RDLLDC in PIDENT RDIS is the entry for coordinating disconnect actions for an intra-RSS (ie, line-to-line) call. Refer to the program listing (Table A) for detailed information.

4.37 For path disconnects involving both RSS and ESS paths, the ESS disconnect programs generally control disconnect actions up to the point in time that a decision is made to tear the call down. The ESS disconnect program, PIDENT DISC, enters RDIS at global RDRCLT to initiate disconnect actions for the RSS portion of the connection. Refer to the RDIS program listing (Table A) for detailed information describing the control sequences. At this time the ESS programs do a restore verify on the channel, RDIS is notified and initiates disconnect actions on the RSS portion of the call. RDIS and DISC then autonomously carry out their respective disconnect actions with DISC handling the ESS portion and RDIS handling the RSS portion of the call.

PIDENT RUTU

4.38 Translation data is required to be resident in the remote terminal for both regular and stand-alone operations. All of the translation data for the RT is also stored in the host ESS although in a different format. PIDENT RUTU provides the control structure necessary to update all of the translation data (in the RT) upon request and to update the data associated with an LREN whenever a recent change on an LREN is made. A request for a ***total*** update can originate from the RT via the data link or from a TTY associated with the host. Two types of translation update requests are allowed: regular and priority. Priority requests are always handled before regular update requests. Recent changes affecting the RSS common block are not transmitted to the RSS, but a total update is requested if the change is to be made immediately effective.

4.39 The RT may not be able to process calls if the translation data base is defective or impaired. The translation update philosophy is such that the RT data base is left intact should the update process abort. RT translation data base integrity requires that the update process be as fast as possible.

4.40 The update control program, PIDENT RUTU, receives a flagged E-level main program entry to global RUTDUD. This is the coordinator state entry which determines the next sequence of actions to be taken. The coordinator states and meanings are as follows:

(a) Idle—No work is underway. Determine the next job to do.

(b) Message confirm—A message has been sent to the RT and the host is waiting (timing) for confirmation.

(c) Send message—The data buffer is loaded with data waiting to be sent. This is a queue state used when the PUC buffer is full.

(d) Time break—Used to take a real-time break during the forming of data in the data buffer where excessive real-time is required.

4.41 Generally, after each acknowledgment from the RT, the coordinator state is set to idle and control is given to RUTDUD. The highest priority task is then executed according to the following rules:

(a) Recent change messages are processed first.

(b) If a priority update request is present, it is processed next with TTY requests taking precedence. However, once a priority update is initialized, it runs to completion with only recent change messages as possible interruptions.

(c) If a regular RSS update is in progress, it can be interrupted by both priority and recent change updates.

For detailed information, refer to the RUTU program listing (Table A).

TRAFFIC

PIDENT RTRF

4.42 PIDENT RTRF is the RSS traffic program, consisting of several routines which perform traffic functions for the RSS feature. RTRF interfaces heavily with the host traffic program and the associated data base. The functions in this program are generally associated with the RSS resources that are controlled by the RSS programs and whose status is kept in the RSS data bases. Refer to the program listing (Table A) for details.

5. ABBREVIATIONS AND ACRONYMS

CC	Central Control
CDPR	Customer Dial Pulse Receiver
CIN	Change in Network
CRA	Call Register Address
CREN	Channel Remote Equipment Number
CS	Call Store
CURL	Call Unexpected Result List
DCNR	RSS Dialing Connection Program
DL	Data Link
ESS	Electronic Switching System
I/O	Input/Output
L-L	Line-to-Line
LREN	Line Remote Equipment Number
MURL	Maintenance Unexpected Result List
OR	Originating Register
POB	Peripheral Order Buffer
PSPD	Permanent Signal Partial Dial
PUC	Peripheral Unit Controller
RCIN	Remote Change in Network
RDIS	RSS Disconnect Program
REN	Remote Equipment Number
RESW	RSS Reswitch Program
RMSG	ROB Loading and Administration Program
RNAD	RSS Network Administration Program

ROB	Remote Order Buffer
ROBE	ROB Execution Program
ROBF	ROB Failure Program
RRTE	RSS Message Routing Program
RSS	Remote Switching System
RSUP	RSS Supervision Report Program
RT	Remote Terminal
RTAD	RSS Terminal Administration Program
RTRF	RSS Traffic Program
SP	Signal Processor
URTA	Unexpected Result Transfer Address
USC	Universal Service Circuit

6. REFERENCES

RSS PIDENTS

(See Table A.)

5 Watt Infrared Flashlight

Overview

This is simple project to convert a Transel–brand (or similar) adjustable zoom flashlight, which should be available at most sporting goods stores, into a very handy infrared (850 nm) flashlight. The infrared flashlight can then be used as an active illuminator for a night vision device, camera, or even for covert signaling.

The project consists of replacing the stock Cree white LED module with a 5 watt, 850 nm infrared LED module in a similar "star" package. The 5 watt infrared module used here is available on eBay for under $10. Search eBay for "1pc 5w 850nm Infrared IR LED 5 Watt" to find many distributors out of China selling these modules. The infrared LED module's overall construction quality isn't very great – I've had the lenses break off – but they'll work well for experimenting, just be careful...

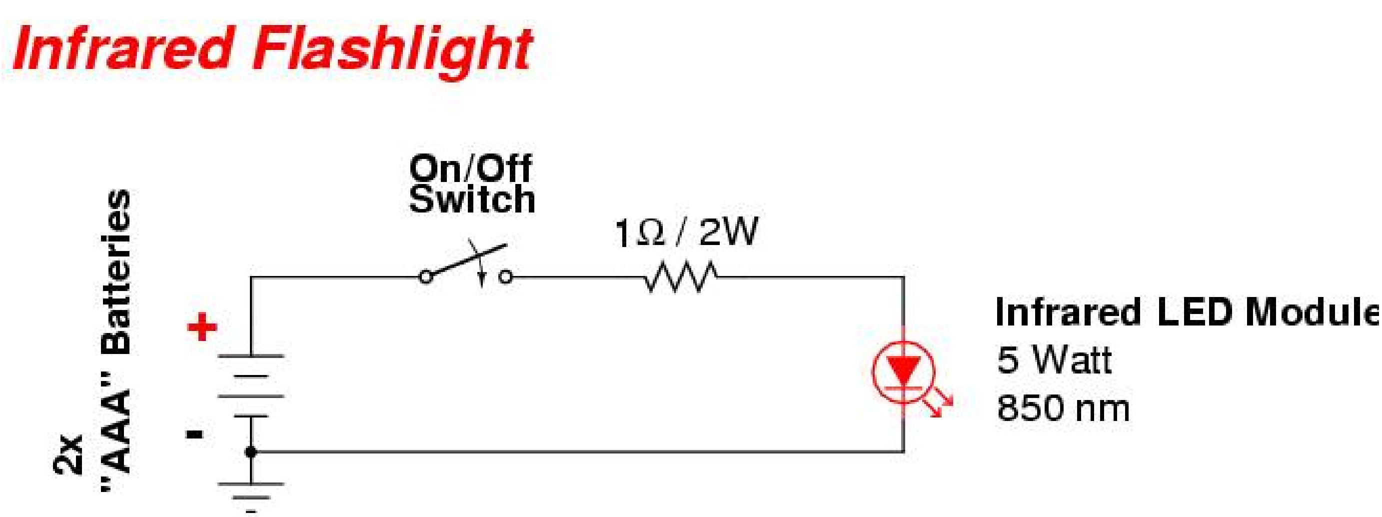
The infrared LED module's specifications are listed as follows:

```
        Emitted Color: 850 nm
        Viewing Angle: 135 degrees
   DC Forward Voltage: 1.4V
   DC Forward Current: 1.4A
Maximum Pulse Voltage: 1.7V
Maximum Pulse Current: 2.0A
```

The stock flashlight ran off of three "AAA" size batteries, which were required for overcoming the white LED's approximate 3.6V forward voltage drop. Since the infrared LED module has a lower voltage requirement, with an approximately 1.4V forward voltage drop, we'll also need to modifiy the flashlight's battery holder to only require two "AAA" batteries.

Using three "AAA" batteries will work (4.5 VDC), but you'd need to increase the value of the dropping resistor accordingly, to around 2.2 ohm / 5 watt.

The stock flashlight also has an internal circuit board for switching between bright, dim, or flashing modes. The components on this board are not really required, so they'll be removed and a single 1 ohm / 2 watt dropping resistor will be added.



That's it!

Pictures & Construction Notes

Overview of the stock adjustable zoom flashlight.

It has words "TRANSEL Connecting the World" on the side.

It requires three "AAA" size batteries (+4.5 VDC).

A similar, usable flashlight appears to be the Duracell Durabeam Ultra 500 series, though I haven't tried it.

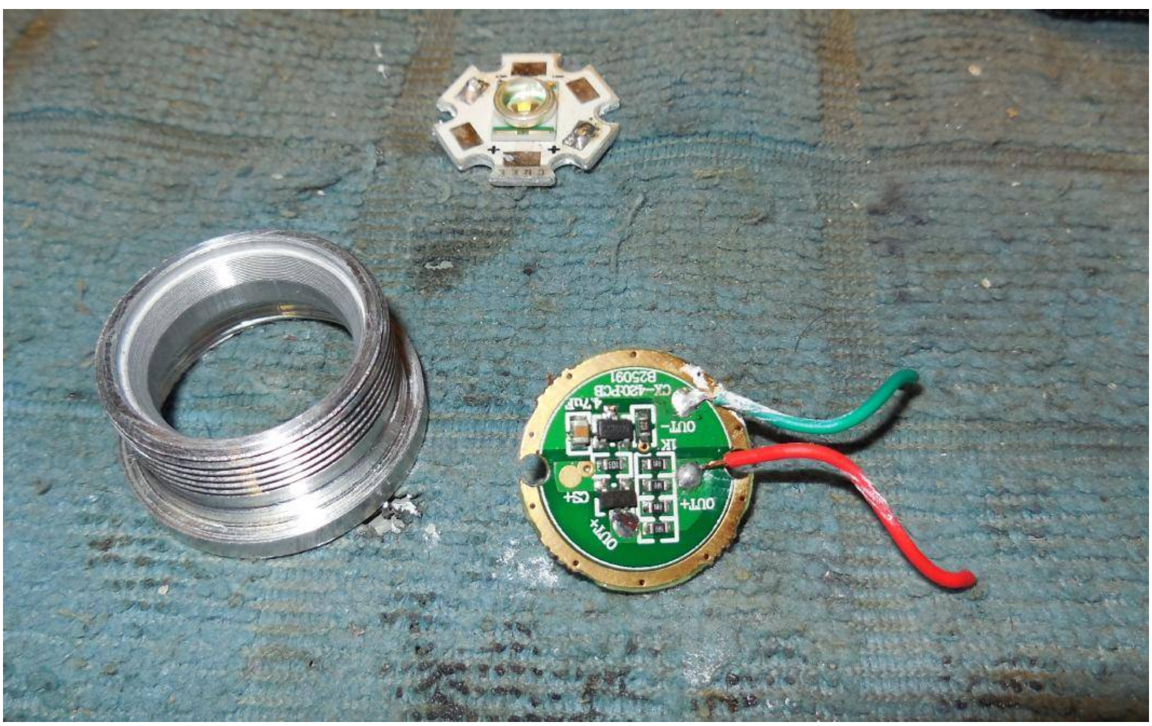
Taking the flashlight apart.

Unscrew the plastic front bezel, then unscrew the locking washer. Use a pair of needle−nose or snap−ring pliers to fit inside the little holes and unscrew the locking washer.

A metal flat washer and a plastic washer are behind the locking washer to secure the LED module against the aluminum holder, which acts as a heatsink.

Closeup of the alumimum body which holds the LED module.

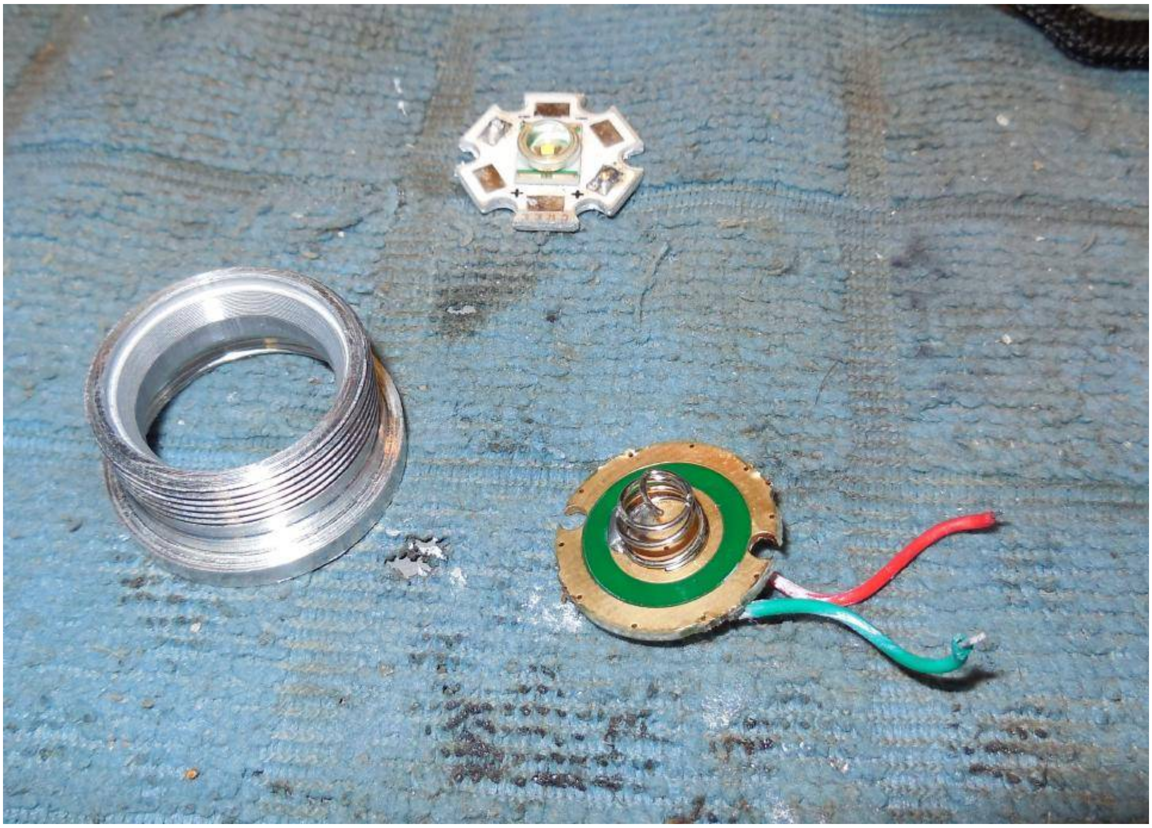
This acts as a heatsink and screws into the body of the flashlight.



Removing the control circuit board for the stock LED module.

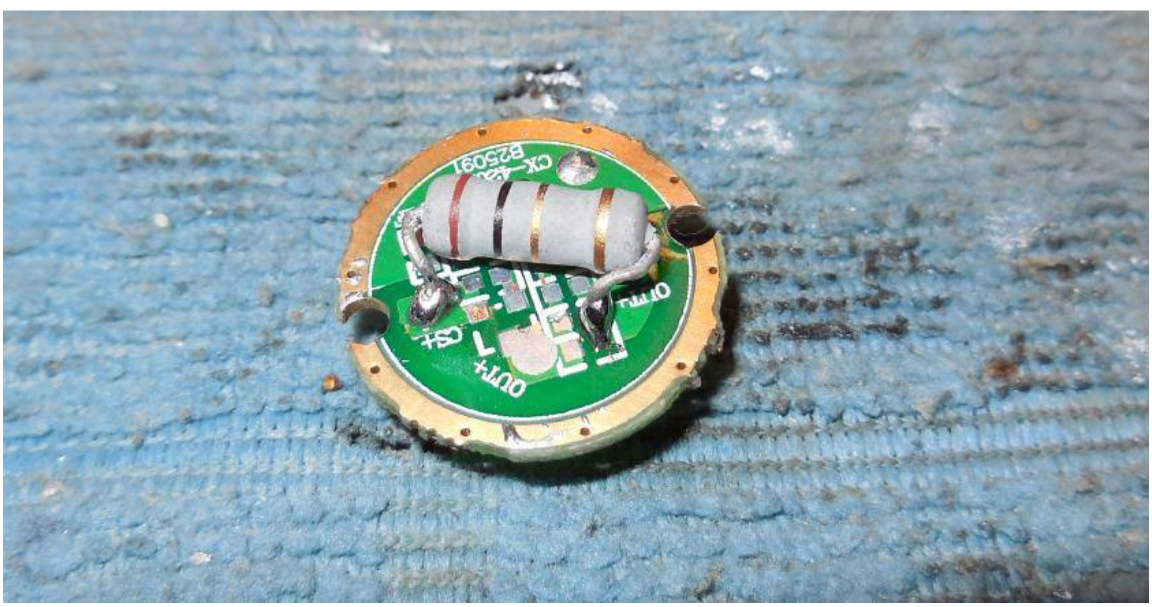
This circuit board has three settings, bright, dim, and flashing, which are selected via the half−step power switch.

The components on this board will not be used for the infrared flashlight and should be removed. Just heat the board with a hot air gun and slide them off.



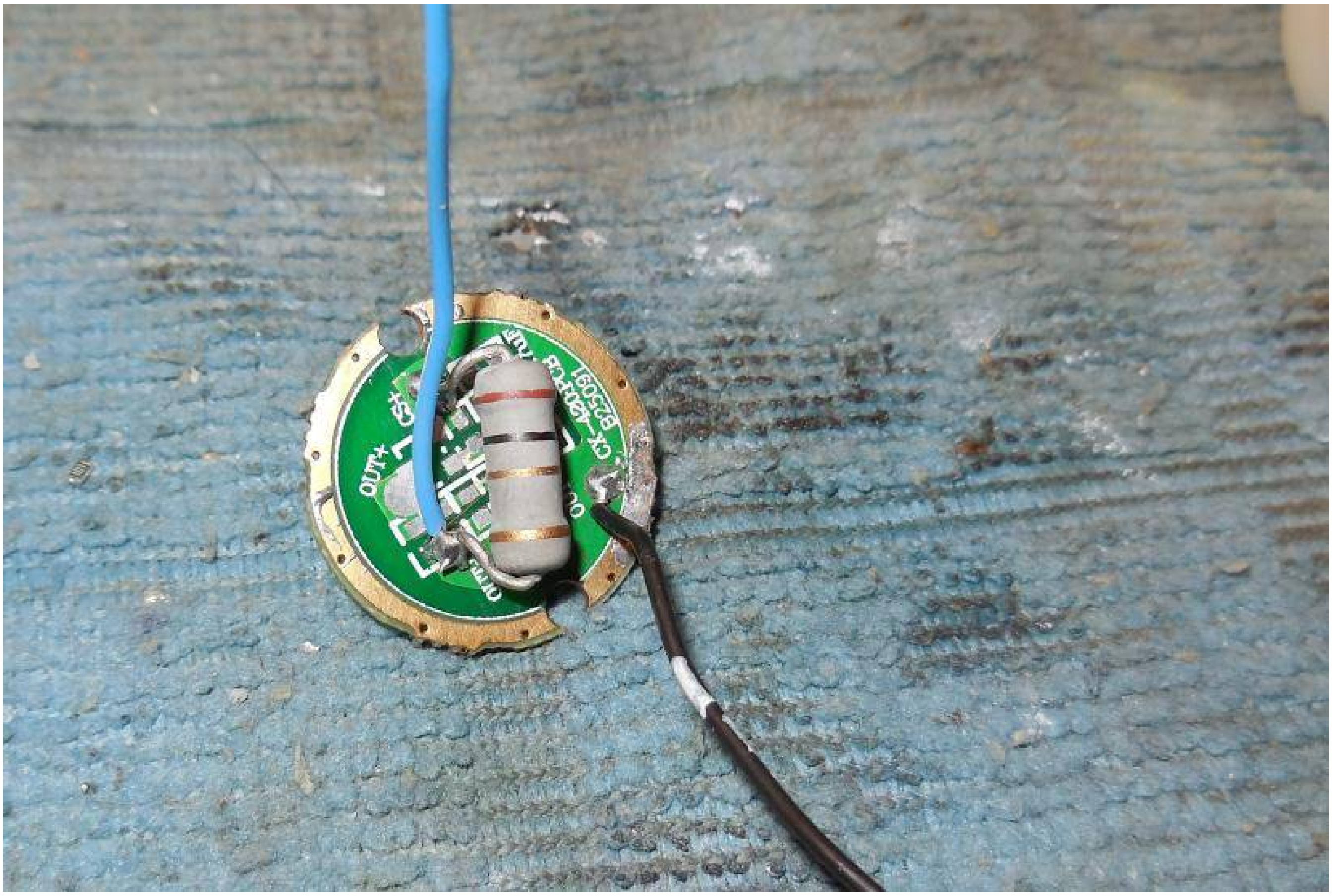
Rear–view of the circuit board.

The center spring is battery positive. The exposed copper ring is battery negative



All the 5 watt infrared LED module requires for operation is a single 1 ohm / 2 watt resistor.

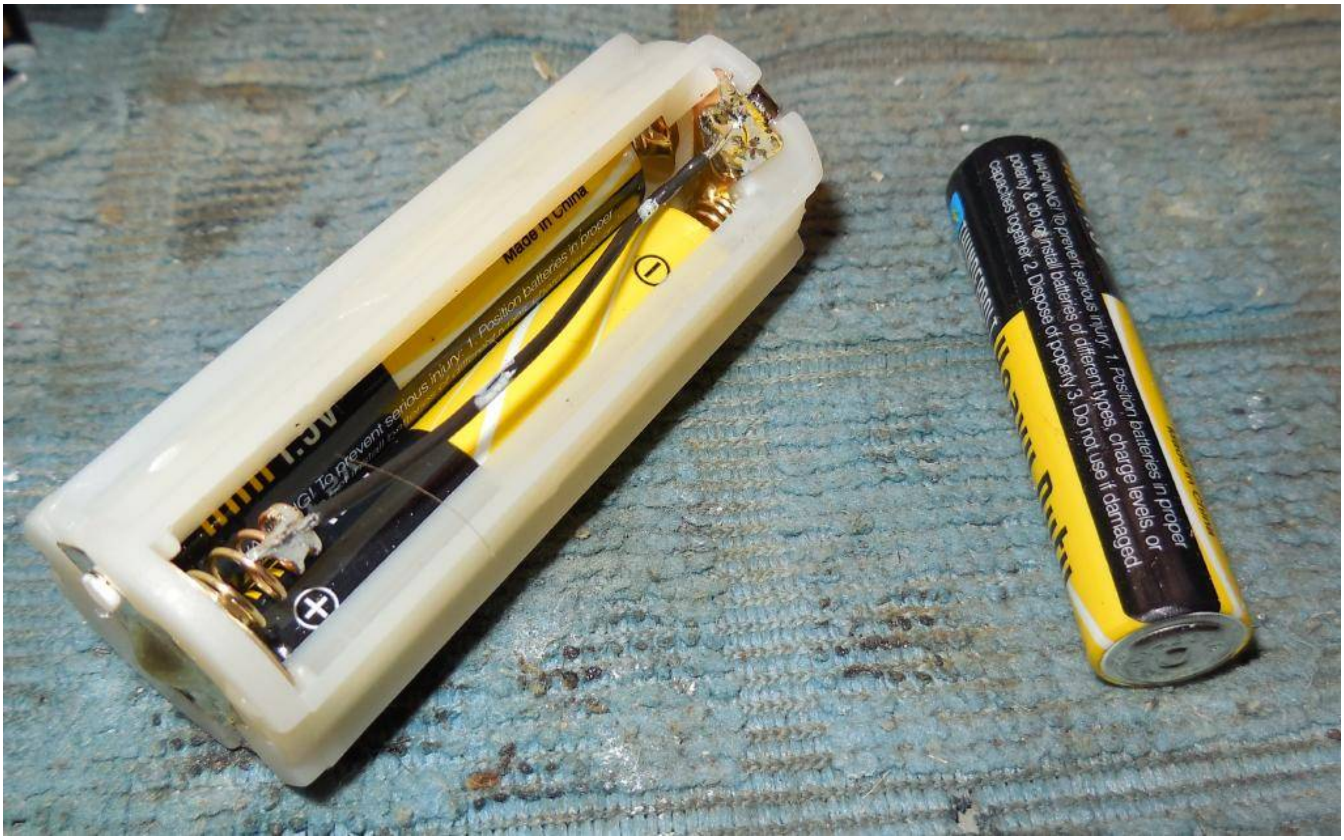
Solder one leg to the CS+ pad and the other leg to the OUT+ pad.



Alternate view, with the connecting wires attached.

The blue wire will go to the infrared module's **+** pad.

The black wire will go to the infrared module's **–** pad.



Modifying the battery holder to only require two "AAA" size batteries.

Remove one of the batteries and replace it by soldering a jumper wire from the spring tab to the flat tab.

This converts the 4.5 VDC battery pack into a 3.0 VDC battery pack.

You'll need to add a dab of heatsink thermal compound to the back of the infrared LED module.

The module should fit flat against the aluminum holder for proper heatsinking.

Be sure the wires are not pinched.

You may need to use a paper punch to enlarge the center hole on the plastic insulator.

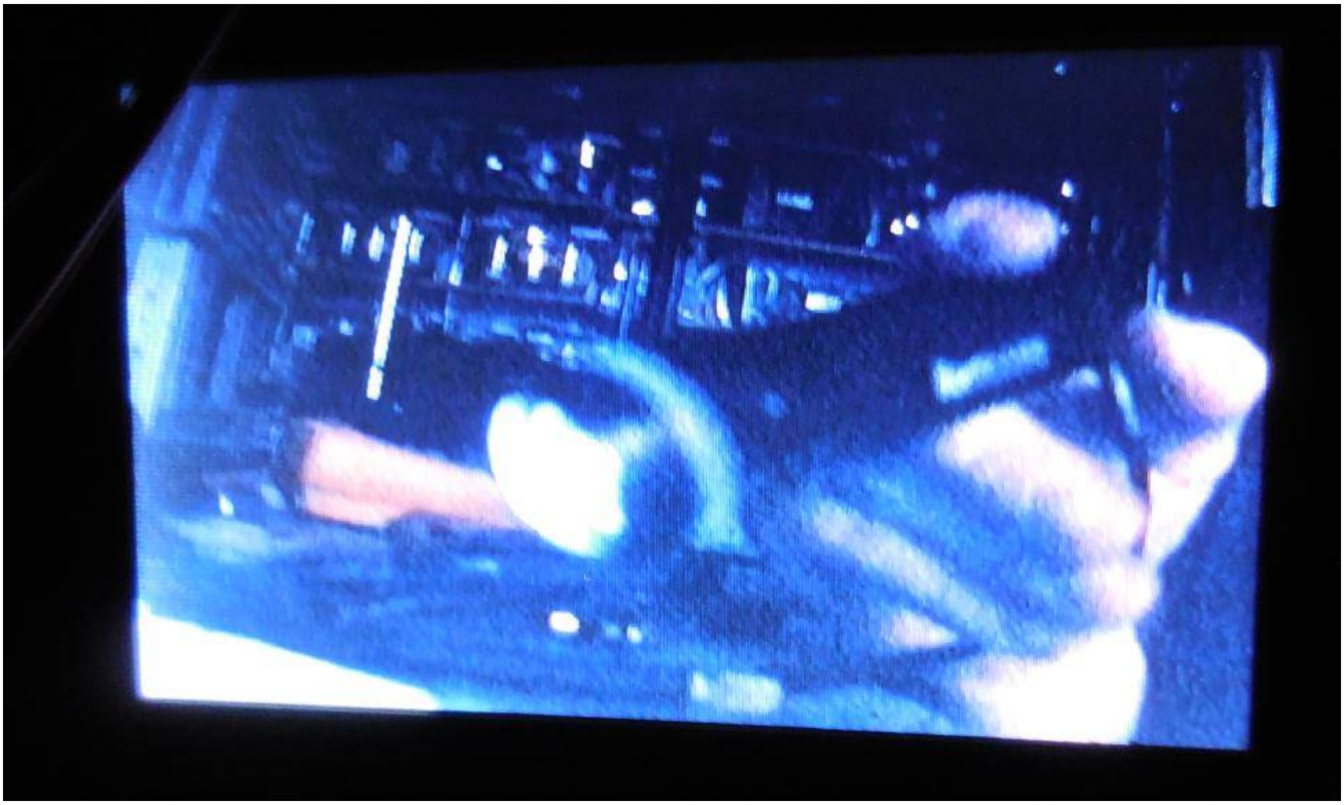
You may also need to enlarge the center hole on the metal compression washer, use a Dremel tool and grinding wheel for this.

This plastic washer is *REQUIRED* to prevent the infrared LED module from shorting out against the locking washer.

Finished 5 watt, 850 nm infrared flashlight module ready to be screwed back into the flashlight's body.

Double−check that the center spring (battery positive) isn't shorted to ground.

Note how the shiny compression washer had its center hole enlarged to fit over the new infrared LED.

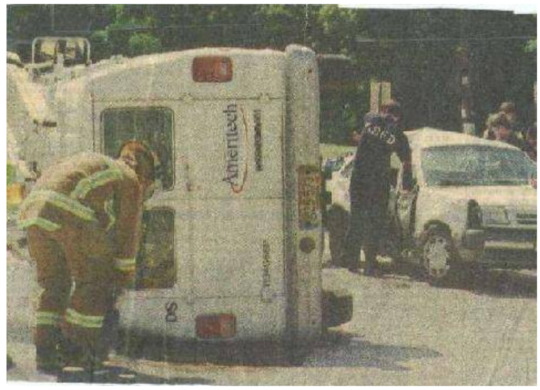


Finished operational 5 watt infrared flashlight as displayed on a TV screen via a black−and−white infrared−compatible video camera.

The final IR flashlight doesn't provide the greatest illumination, but the adjustable "zoom" lens allows you to concentrate the IR energy into a specific area, which is something most flashlights don't allow you to do.

Bonus

End of Issue #116



Any Questions?

Editorial and Rants

Nelson Mandela Was a Communist Terrorist Backed by Zionists

December 6, 2013 – *From:* blacklistednews.com

by Lee Rogers

Nelson Mandela, the former President of South Africa, has passed away at the age of 95. The big American media outlets are currently spending hours upon hours of air time praising Mandela as some sort of angelic icon of peace. This is a total fabrication of reality. Mandela originally aligned himself with the African National Congress, a Communist revolutionary group heavily influenced and financed by Zionist Jews.

This organization would be responsible for all sorts of atrocities in South Africa which eventually led to Mandela's time in jail. Mandela co-founded the militant wing of the ANC with various South African Communists including an Israeli Jew by the name of Arthur Goldreich. The group was called *Umkhonto we Sizwe* or 'Spear of the Nation.' It is important to note that the ANC not only attacked official government buildings, but even non-government targets like movie theatres as well. It was this activity that made it easy for several countries, including the United States, to label the ANC as a terrorist organization. Mandela himself was even on the U.S. terrorist watch list until 2008. Fun facts about Mandela such as these are completely ignored by all of the big media outlets because it runs contrary to the portrait they are trying to paint.

In order to understand who Mandela really was, it is necessary to understand the real history of apartheid South Africa. The policies of apartheid, or racial segregation, were largely implemented starting in the late 1940s to early 1950s. These policies were originally intended to give the different races within South Africa an independent area of their own. It was argued by South African leaders at the time that South Africa wasn't a single nation but was made up of several different racial groups which should be split apart. Although the merits of these policies or lack thereof could be argued, the policies were not as nefarious as we have been led to believe.

The so-called apartheid South African government which was dominated by White Europeans had made South Africa a successful independent first-world nation. This was the real reason why Jewish Communists sought to use the ANC as a way to demonize the White European leaders in power. South Africa represented an independent economic and military power that needed to be brought under their influence.

Much like the phony civil rights movement in America, the ANC was dominated by Jewish Communists even up until the 1990s when Mandela took power. A recent article from *Haaretz* notes the following:

"The African National Congress, the liberation movement that became the governing party in 1994, also had a full complement of Jews, including Joe Slovo, Ronnie Kasrils, and Denis Goldberg."

The ANC would not only engage in acts of terror against Whites, but they would also do the same against Blacks who they suspected of collaborating with Whites. Specifically they would execute and torture people by igniting a rubber tire filled with petrol that they forced over their chests and arms. The practice referred to as 'necklacing' would typically take the victim over 15 minutes to die in certain cases. Hundreds of executions using this method were carried out by the ANC. Even Mandela's one-time wife Winnie would implicitly endorse this method of torture and execution.

Early in his life, Mandela was surrounded by Jews and was given his first job as a clerk by a Jewish lawyer named Lazar Sidelsky. He would associate himself with a large number of Jewish Communists including some of the ones mentioned previously. In fact during a sweep of the ANC in the early 1960s which resulted in his arrest and lengthy prison sentence, a significant number of Jews were also arrested. Enormous caches of weapons and explosives held by the ANC were also uncovered. A recently published article from *Tablet Magazine* goes into great detail about how Mandela was aligned strongly with a significant number of Zionist and Communist Jews before and up until the time of his arrest.

In 1985, the President of South Africa Pieter W. Botha offered to release Mandela from prison if he would unconditionally reject violence as a political instrument. Mandela refused the offer. This fact completely destroys the notion that Mandela was a man of peace.

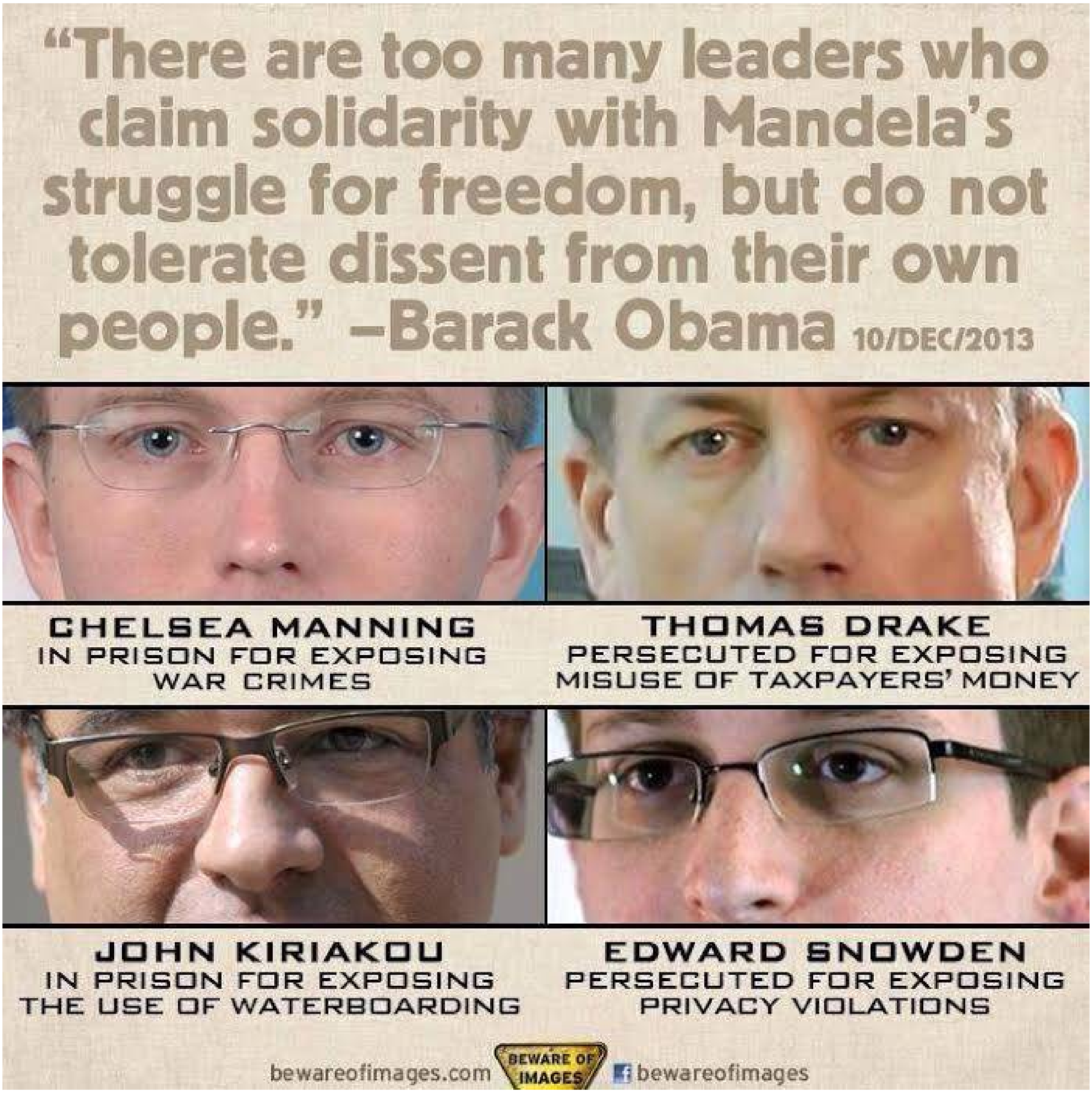
Mandela's release from prison in 1990 was greeted with widespread media coverage from all of the major Jewish−controlled press outlets, including American mainstream media. Instead of focusing on his past, he was portrayed as a man of peace and an iconic freedom fighter. The biased media spin was used to make people forget about who he really was. Amazingly the ridiculous media extravaganza helped Mandela become President of South Africa allowing his Jewish backers to change South Africa into a nation run by Communist principles.

Since Mandela's ascent to the Presidency, the South African economy has actually worsened compared to when it was led by the apartheid government. A *BBC* article goes into great detail about how many things were better before the ANC and Mandela took power. In fact, economic inequality is far worse now than before. Unemployment and poverty is rampant with many South Africans living in shacks. Dissent against the ANC is largely stifled as one would expect in a nation run by Communist principles. There has also been a substantial increase in the number of White South Africans murdered by Blacks since this transformation. Specifically, White South African farmers have been primary targets during this reign of murder and terror.

Simply put, Mandela was nothing more than a cult of personality fraud who has brought ruin to South Africa. Many Black South Africans are actually worse off now than under the alleged evils of the apartheid government. He was always a puppet for powerful Jewish interests who were the ones that really helped him gain power in South Africa. It is no wonder why Barack Obama had so many kind things to say about Mandela because Obama is literally doing the same thing to America that Mandela did to South Africa. Mandela should be remembered with disdain and not with reverence.

The only good news is that Obama disrespected Mandela's funeral with his narcissistic comedy routine we call a presidency:

Change!

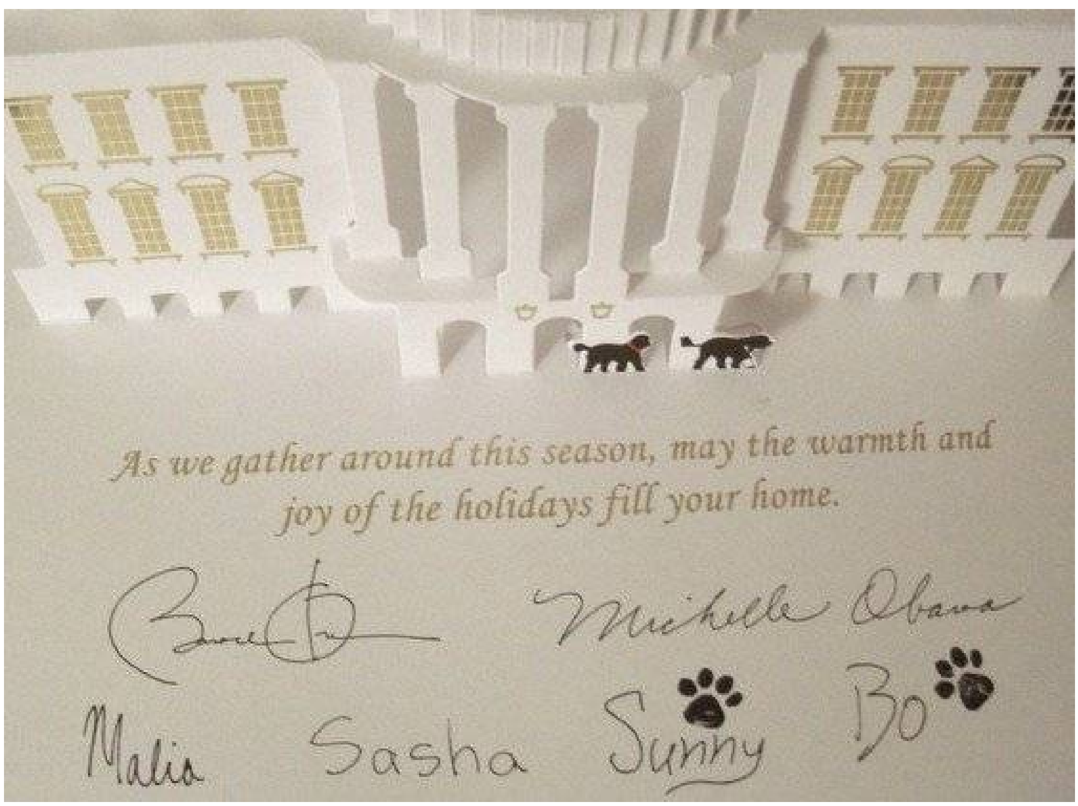


"In the end the Party would announce that two and two made five, and you would have to believe it. It was inevitable that they should make that claim sooner or later: the logic of their position demanded it. Not merely the validity of experience, but the very existence of external reality was tacitly denied by their philosophy."

––– Quote from *1984*, by George Orwell.

Oh no! The Communists and Jean−Luc Mélenchon are gaining support in France! The TV media said so, so it must be true...

Nope! Just more of that "non−existent" liberal bias in the mainstream media. Above is a picture taken by Dutch journalist Stefan de Vries − who just happened to live in the area.



Barack Hussein Obama's Christmas card – with no mention of Christmas or Christ/Jesus – just as you'd expect from a Marxist Muslim.

His monkey paw prints are a cute little touch, though...