JVCKENWOOD

Built-in Board Interface (Digital Encryption)

Ver 1.10

Version:	1.10
Last Updated:	2014.11.21
Language:	English

Revision

Date	Description
2012.05.24	Initial Version 1.00
2014.11.21	Update for NX-5000 series interface Added new commands for AM Calculate, SUID Report and Multi-stream. Added a new sheet for the KWD-AE30/AE31 board size. Revised to Version 1.10

Table of Contents

Figures

About this Technical Document

This document describes the interface for connecting Digital Board such as Encryption. This document is intended to provide technical elements.

In this document, the specifications necessary to connect a Digital board to the NEXEDGE radio.

JVC Kenwood Corporation (hereinafter referred to as "JVC Kenwood") does not warrant connectivity or quality of communications by means of this document.

This document is subject to change without notice due to interface additions or modifications.

1. Glossary

Following are the definitions of the terms used in this document.

1.1 Terms

NEXEDGE	This is a brand name of Kenwood two way radio
AES	Advanced Encryption Standard
DES	Data Encryption Standard
Board	This is intended the SCM board
KVL-3000	Key Variable Loader produced by Motorola product
Zeroize	To clear all Encryption Keys
IV	Initialization Vector, the starting point of the encryption algorithm for each transmission
KVL	Key Variable Loader
OFB	Output Feed Back, one operating mode for encryption
ECB	Electronic Code Book, one operating mode for encryption
ALGID	Algorithm ID to indicate the type of encryption algorithm
KID	Key Identifier to indicate the encryption key for the message

1.2 Abbriviation

CH	Channel
ESN	Electrical Serial Number
ID	Identity
IV	Initialization Vector
KVL	Key Variable Loader
SCM	Secure Cryptographic Module
CKR	Common Key Reference
OFB	Output Feed Back
ECB	Electronic Code Book
ALGID	Algorithm ID
KID	Key Identifier
SID	Stream Identifier

2. Reference

This documentation is referred the following specification.

[1] TIA-102.AACD PROJECT 25 DIGITAL LAND MOBILE RADIO - KEY FILL DEVICE (KFD) INTERFACE PROTOCOL

3. Procedure

3.1 Power-up

The Board must follow the power-up sequence as below drawing.

If the Board was not cooperate the sequence, the radio will be suspend in the faulty mode.

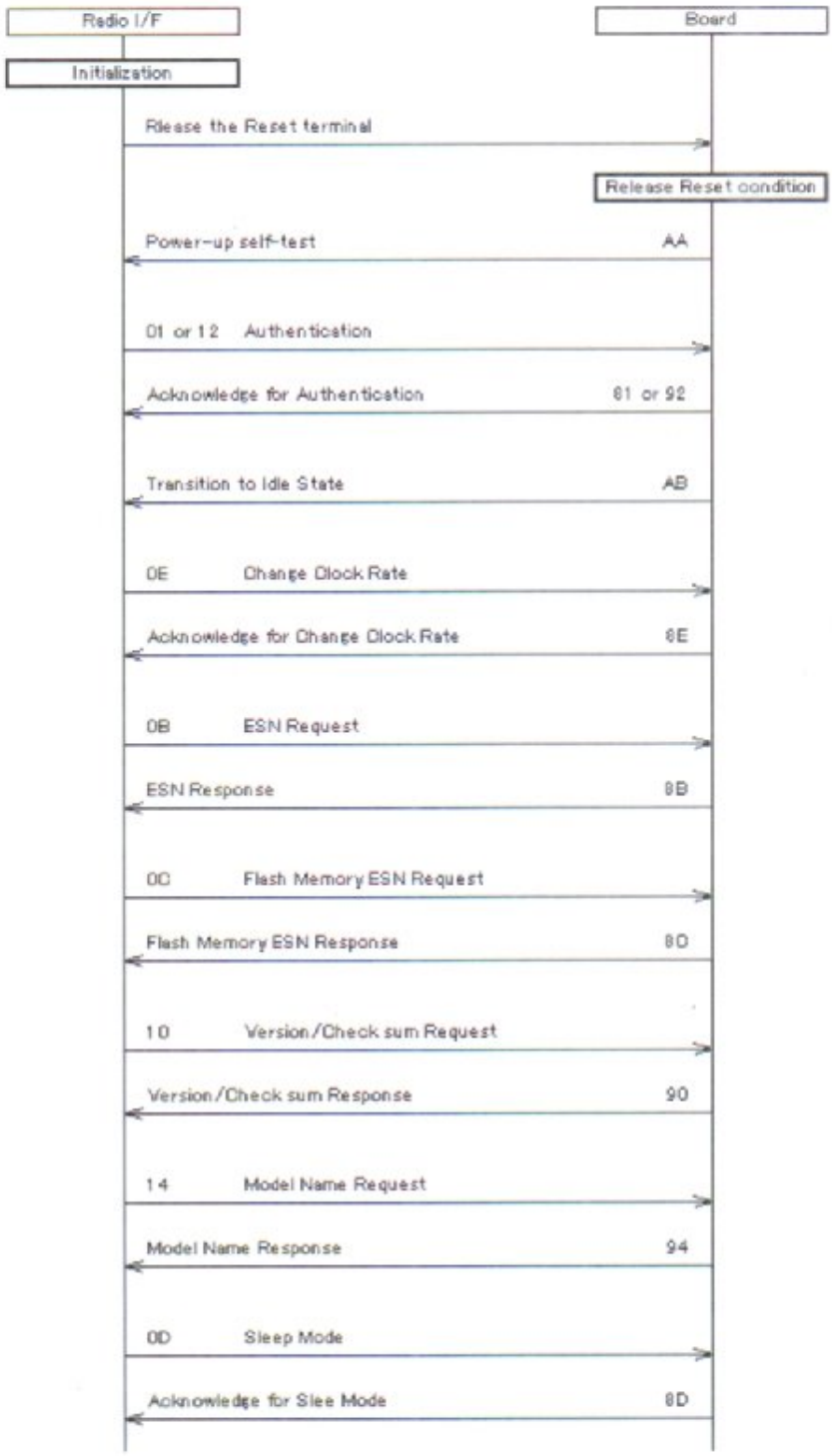


Fig. 3-1 Power-up

3.2 Encryption

3.2.1 Encryption

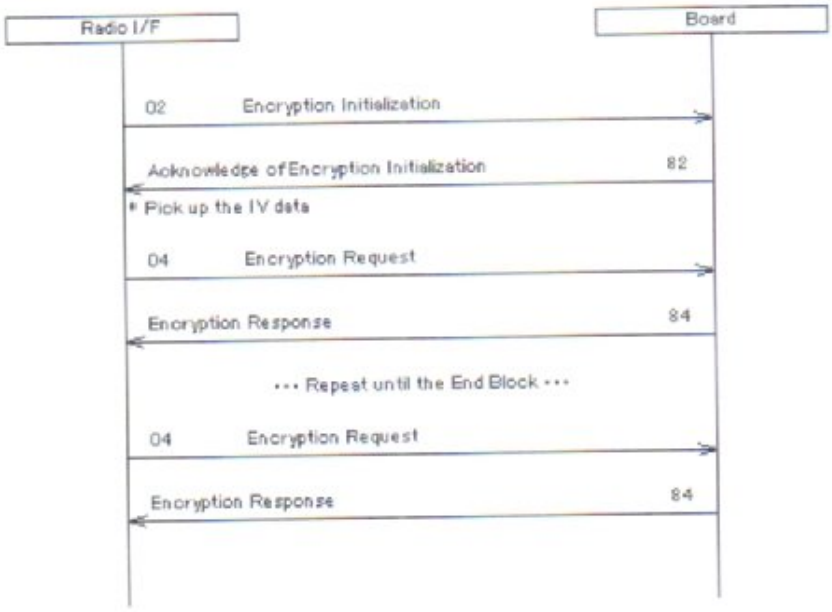


Fig. 3-2 Encryption

3.2.2 Decryption

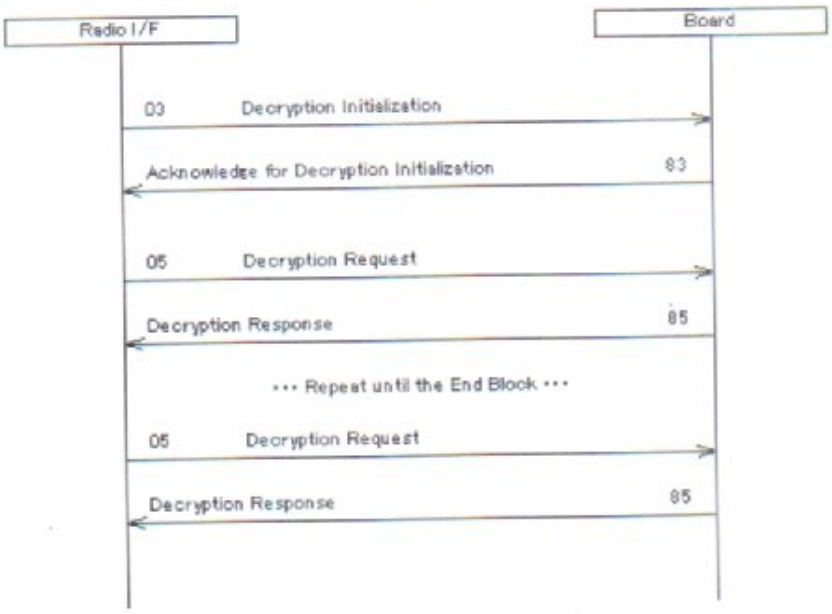


Fig. 3-3 Decryption

3.2.3 Zeroize

This sequence will appear when the radio clear the Key data in the Board.

Fig. 3-4 Zeroize

3.2.4 KVL Mode

This sequence will appear for communicating with KVL equipment.

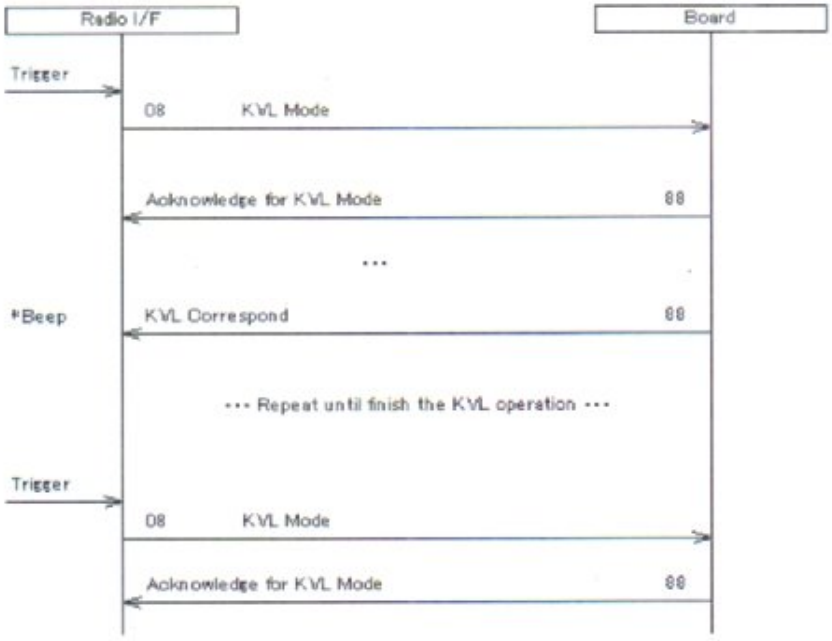


Fig. 3-5 KVL Mode

The radio will emit a beep sound when received the "88" command from the Board for recognize an action.

3.2.5 Sleep Mode

This sequence will appear when the radio states idle or power save to conserve the current consumption.

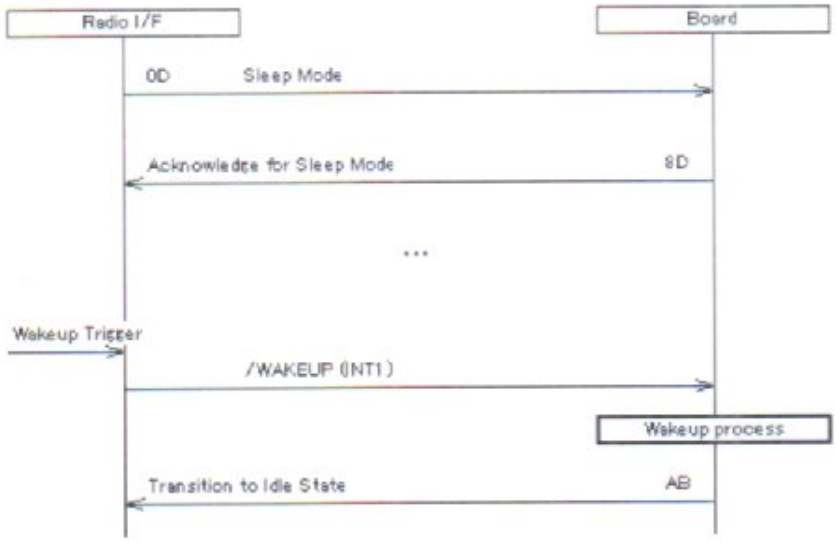


Fig. 3-6 Sleep Mode

3.2.6 Beat Shift

This sequence will appear when the selected channel has configured the Beat Shift function.

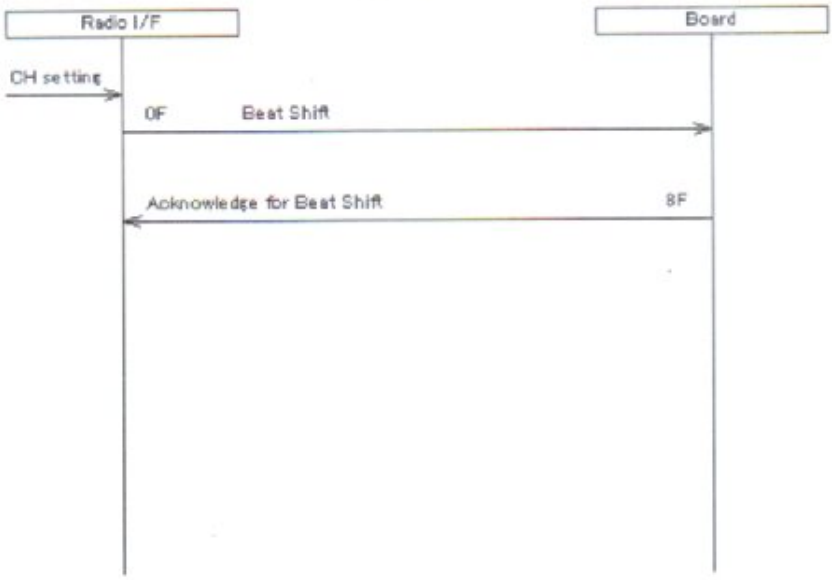


Fig. 3-7 Beat Shift

4. Command

4.1 Connection

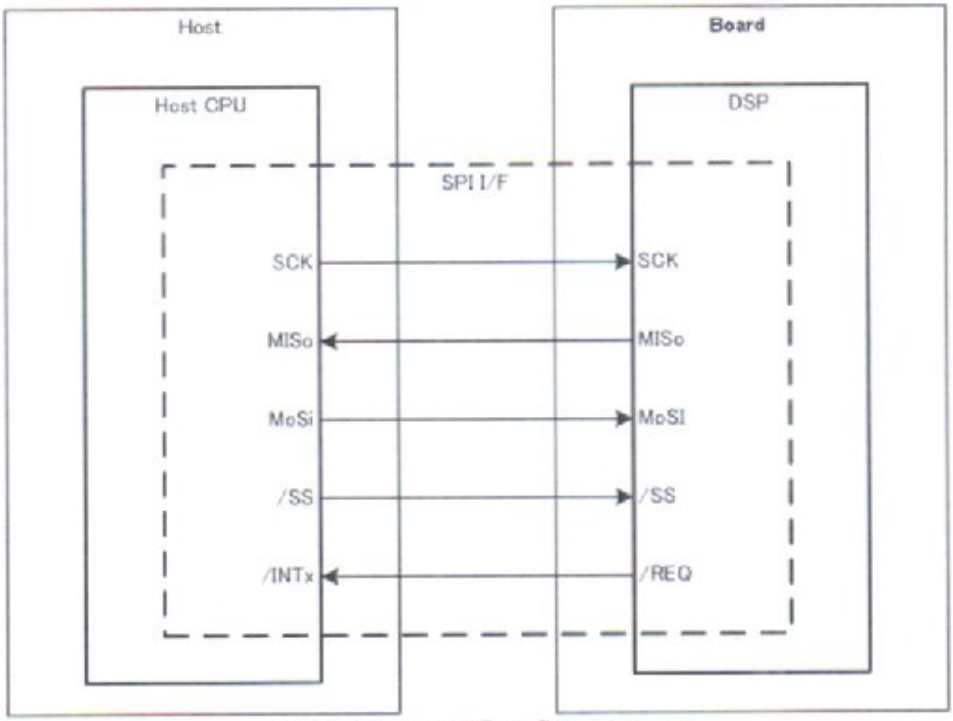


Fig. 4-1 SPI Interface HW Configuration

4.2 Interface Specification

4.2.1 Command Format

MSB LSB

Command (8bit)	Data Length n(8bit)	Data 1	Data 2	...	Data n
1 WORD		2 WORD	3 WORD	...	(n+1) WORD

Fig. 4-2 Command Format

Remarks

- Data send/received by the SPI are to be MSB first, and 16 bit units.
- Of the first 16 bit's 8 bit is command, and the next 8 bit is the amount of data word followings, followed by the data.
- For data longer than 16 bits, packets will be added, and data will be send.
- For command only data, the data length will be zero.

4.2.2 Access Method

In order to send commands to the Board from the Host CPU, data will be sent by normal SPI format, and data will be received from the Receive Interrupt of the Receive Register.

In order to send data from the Board to the Host CPU, toggle the /REQ, which is configured as the general purpose output port, with active low, and make an interrupt to the Host CPU. After the Host CPU receives interrupt, it will send a Data Load command from the SPI, and receive command and data from the Board.

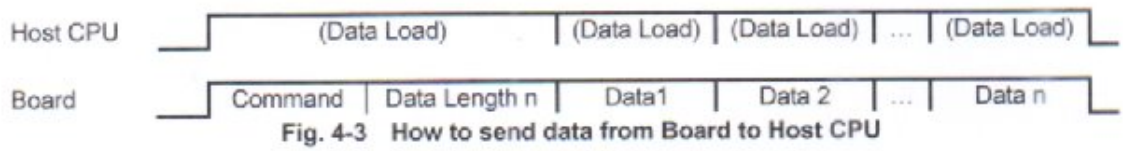


Fig. 4-3 How to send data from Board to Host CPU

4.3 Command List

Command	Function	Alias Name	Direction
AA	Power Up Self Test	ACK_SELF_TEST	<-
AB	Transition to Idle State	ACK_IDLE	<-
00	Data Load	DATA_LOAD	->
01	Authentication Command 1	AUTH_CMD1	->
81	Authentication Command Ack 1	ACK_AUTH1	<-
02	Encryption Initialize Command	ENC_INIT_CMD	->
82	Encryption Initialize Command Ack	ACK_ENC_INIT	<-
03	Decryption Initialize Command	DEC_INIT_CMD	->
83	Decryption Initialize Command Ack	ACK_DEC_INIT	<-
04	Encryption Command	ENC_CMD	->
84	Encryption Command Ack	ACK_ENC	<-
05	Decryption Command	DEC_CMD	->
85	Decryption Command Ack	ACK_DEC	<-
06	Zeroize Command	ZERO_CMD	->
86	Zeroize Command Ack	ACK_ZERO	<-
87	Zeroize (RAM) Command Ack	ACK_ZERO_RAM	<-
08	KVL Mode Command	KVL_MOD_CMD	->
88	KVL Mode Command Ack	ACK_KVL_MOD	<-
09	Status Request	STS_REQ	->
89	Status Response	STS_RSP	<-
0A	Firmware Update Command	FIRM_UPDT_CMD	->
8A	Firmware Update Command Ack	ACK_FIRM_UPDT	<-
0B	ESN Request	ESN_REQ	->
8B	ESN Response	ESN_RSP	<-
0C	Flash Memory ESN Request	FLSH_ESN_REQ	->
8C	Flash Memory ESN Response	FLSH_ESN_RSP	<-
0D	Sleep Mode Command	SLEEP_CMD	->
8D	Sleep Mode Command Ack	ACK_SLEEP	<-
0E	Change Clock Rate Command	CHG_CLK_CMD	->
8E	Change Clock Rate Command Ack	ACK_CHG_CLK	<-
0F	Beat Shift Command	BSHIFT_CMD	->
8F	Beat Shift Command Ack	ACK_BSHIFT	<-
10	Version/Checksum Request	VER_SUM_REQ	->
90	Version/Checksum Response	VER_SUM_RSP	<-
12	Authentication Command 2	AUTH_CMD2	->
92	Authentication Command Ack 2	ACK_AUTH2	<-
13*	SKL Mode Command	SKL_MOD_CMD	->
93*	SKL Mode Command Ack	ACK_SCK_MOD	<-
14	Model Name Request	MODEL_REQ	->
94	Model Name Request Ack	MODEL_RSP	<-
15	AM Calculate Command	AM_CALC_CMD	->
95	AM Calculate Command Ack	ACK_AM_CALC	<-
16	SUID Report Command	SUID_RPT_CMD	->
96	SUID Report Command Ack	ACK_SUID_RPT	<-
17*	Encryption Initialize MSV Command	ENC_INIT_MSV_CMD	->
97*	Encryption Initialize MSV Command Ack	ACK_ENC_INIT_MSV	<-
18*	Decryption Initialize MSV Command	DEC_INIT_MSV_CMD	->
98*	Decryption Initialize MSV Command Ack	ACK_DEC_INIT_MSV	<-
19*	Encryption MSV Command	ENC_MSV_CMD	->
99*	Encryption MSV Command Ack	ACK_ENC_MSV	<-
1A*	Decryption MSV Command	DEC_MSV_CMD	->
9A*	Decryption MSV Command Ack	ACK_DEC_MSV	<-
1B*	Status Request MSV	STS_REQ_MSV	->
9B*	Status Response MSV	STS_RSP_MSV	<-
60-6F	OTAR KMM Command	OTAR_KMM_CMD	->
E0-EF	OTAR KMM Command Ack	ACK_OTAR_KMM	<-
FF	Invalid Response	INVALID_RSP	<-

Remarks

- Direction. [->] : Host CPU to Board. [<-] : Board to Host CPU
- *: The command is available for KWD-AE31/DE31 only.

4.4 Session Details

4.4.1 Power Up Self Test

This command is provided to output the result of Power-Up Self-tests to the Host CPU after reset. If the Self Test Result is 0000, Self-Tests succeeded, and the module will transit to the next state. If the data is anything except for 0000, at least one of the self-test has failed, and the module will transit to an error state. For the details of Self Test Result's data format, refer to the Status Request item later defined.

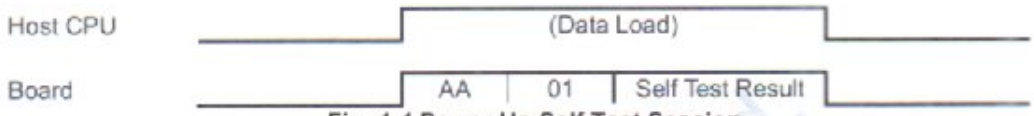


Fig. 4-4 Power Up Self Test Session

4.4.2 Authentication

Perform ESN authentication of the Host CPU. After the Power Up Self Test resulting success, the module will wait for AUTH_CMD1 or AUTH_CMD2 for about 10 seconds. If the received ESN matches the stored value, return a success (0000) and transit to the next state. If the received ESN does not match, zeroize all keys, return a failure (FFFF) and transit to the next state. If either AUTH_CMD1 or AUTH_CMD2 is not received during the waiting period, zeroize all keys, return Zeroize Command Ack (86), and transit to an error state.

When the Host sends AUTH_CMD1, the infinite flag is set (i.e., the active tamper mechanism is for only RAM stored keys), and when AUTH_CMD2 is sent, the infinite flag is not set (i.e., the active tamper mechanism is for both RAM and EEPROM stored keys).

Fig. 4-5 Authentication Session (authentication success or failure)

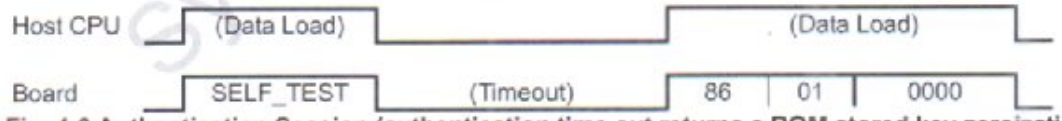


Fig. 4-6 Authentication Session (authentication time out returns a ROM stored key zeroization ACK)

4.4.3 Encryption Initialization

Encryption initialization process. Any encryption shall take place after this session. Key expansion, CRNG tests are performed, and after initialization, Algorithm ID, Key ID, and IV (MI) is returned according to the CKR. Encryption can be initialized by either specifying the CKR or specifying both the Key ID and Algorithm ID.

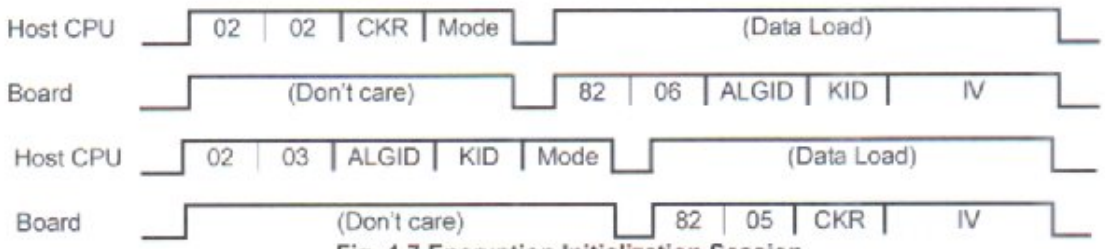


Fig. 4-7 Encryption Initialization Session

CKR : 0001 to FFFF
Mode : OFB Mode (0000), ECB Mode (0001)
ALGID : DES (0081), AES-256 (0084), AES-128 (0085)
KID : 0001 to 1000
IV : Any 4 word except for all 0

- If an invalid mode and/or algorithm ID is assigned, an INVALID_RSP is returned.
- If an invalid CKR (i.e., there are no such keys with the CKR) is assigned, an INVALID_RSP is returned.
- If the CRNG test fails, an INVALID_RSP is returned, and the module transits to an error state.

4.4.4 Decryption Initialization

Decryption initialization process. Any decryption shall take place after this session. The host will send information with the following order; Algorithm ID, Key ID, Mode, IV. After initialization including key expansion, return the CKR.

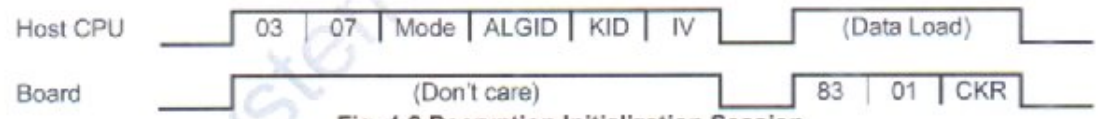


Fig. 4-8 Decryption Initialization Session

CKR : 0001 to FFFF
Mode : OFB Mode (0000), ECB Mode (0001)
ALGID : DES (0081), AES -256(0084), AES-128 (0085)
KID : 0001 to 1000
IV : Any 4 word except for all 0

- If an invalid mode and/or algorithm ID is assigned, and INVALID_RSP is returned.
- If an invalid algorithm ID / Key ID (i.e., there are no such keys with the algorithm ID / Key ID) pair is assigned, an INVALID_RSP is returned.

4.4.5 Encryption

Encryption. A ciphertext is output with the given plaintext. Initialization shall take place before this function. After the command, the Host CPU adds the plaintext required to encrypt. DES is 4 Word and AES is 8 Word. After encryption, the ciphertext is returned. If initialization is not performed, an INVALID_RSP is returned.

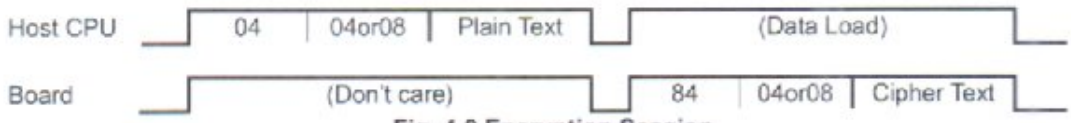


Fig. 4-9 Encryption Session

4.4.6 Decryption

Decryption. A plaintext is output with the given ciphertext. Initialization shall take place before this function. After the command, the Host CPU adds the ciphertext required to decrypt. DES is 4 Word and AES is 8 Word. After decryption, the plaintext is returned. If initialization is not performed, an INVALID_RSP is returned.

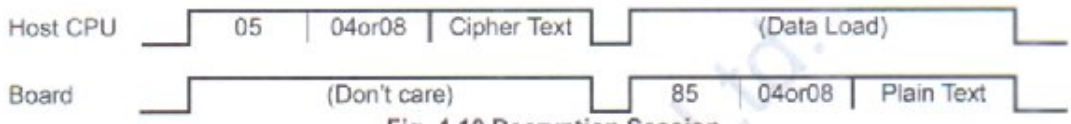


Fig. 4-10 Decryption Session

4.4.7 Zeroize

Zeroization of the keys assigned by their Active Keyset's CKR. If the data length is 99, all keys will be zeroized. After zeroization, ACK_ZERO is returned. If the infinite flag is not set and a tamper is discovered, the module will zeroizes all keys and output ACK_ZERO to the host CPU (for the status, refer to Fig. 3-4 Zeroize).

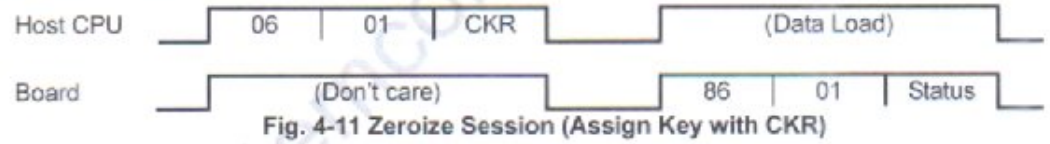


Fig. 4-11 Zeroize Session (Assign Key with CKR)

Fig. 4-12 Zeroize Session (for All keys)

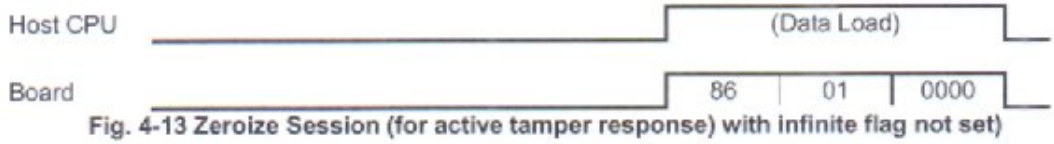


Fig. 4-13 Zeroize Session (for active tamper response) with infinite flag not set)

4.4.8 Zeroize (for RAM stored keys)

When the infinite flag is set and an active tamper is discovered, RAM stored keys will be zeroized, and the module will output an ACK_ZERO_RAM.

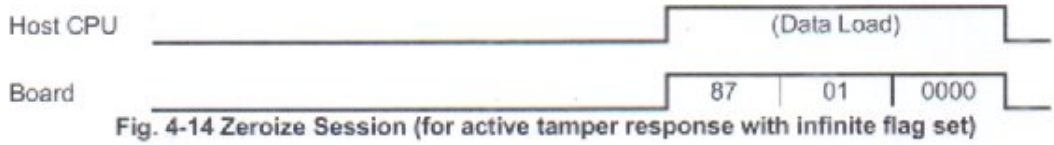


Fig. 4-14 Zeroize Session (for active tamper response with infinite flag set)

4.4.9 KVL Mode

A session to communicate with Key Loader (example: KVL-3000/3000plus/4000).

KVL Mode On (0001) : Cease all cryptographic operation, and transit to the communication mode.

KVL Mode Off (0000) : Cease communication with KVL. Enc/decryption shall be initialized before use.

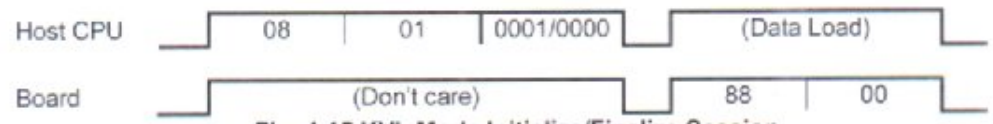


Fig. 4-15 KVL Mode Initialize/Finalize Session

When the session ends with Key Loader: The following data will be sent to the Host CPU.

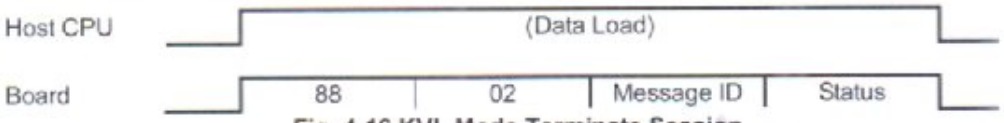


Fig. 4-16 KVL Mode Terminate Session

Status

Code	Status	Circumstance
0000	Command was performed	• The command was successfully performed.
0001	Command could not be performed	• Invalid ALGID • Invalid Key Length • Keyset ID and Key Format do not match • Invalid Parity (DES) • Other errors
0002	Item does not exist	• If the Keyset does not exist. • When an invalid key is to be zeroized. • When a SLD is invalid to the Keyset ID.
0003	Invalid Message ID	• When the Message ID is invalid.
0005	Out of memory	• When the memory is full.
0006	Could not decrypt the message	• If the KMM is encrypted.
0009	Invalid Algorithm ID	• Algorithm ID is not valid or present
000B	Module Failure	• Encryption Hardware failure
000E	Invalid WACN or System ID	• WACN ID or System ID not present
000F	Invalid Subscriber ID	• Subscriber ID not valid or present

Fig. 4-17 Status Details

4.4.10 Status Request

Request the status of the module. During error state, the module will only accept this command. The Self-Test Status will be utilized by the Power Up Self Test session, discussed earlier.

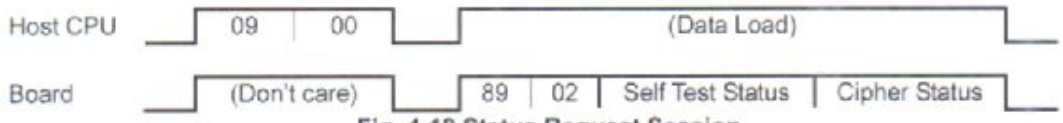


Fig. 4-18 Status Request Session

Self Test Status

Bit	Function	0	1
0	Power Up Selft Test	Completed	Testing
1	Firmware Integrity Test	Success	Error
2	DES Encryption KAT	Success	Error
3	AES Encryption KAT	Success	Error
4	DES Decryption KAT	Success	Error
5	AES Decryption KAT	Success	Error
6	LFSR KAT	Success	Error
7	SHA-256 KAT	Success	Error
8	CMAC KAT	Success	Error
9	Firmware Update	Success	Error
10	Continuous RNG Test	Completed	Testing
11	Continuous RNG Test Result	Success	Error
12	(reserved)	Always 0	
13	(reserved)	Always 0	
14	(reserved)	Always 0	
15	(reserved)	Always 0	

Fig. 4-19 Self Test Status Data Format

Cipher Status

Bit	Function	0	1
0	FIPS Mode	Not validated mode	Validated mode
1	(reserved)	Always 0	
2	Encryption Algorithm	DES	AES
3	Decryption Algorithm	DES	AES
4	(reserved)	Always 0	
5	Encryption Mode	00 : ECB	01 : OFB
6		10 : CBC	11 : (reserved)
7	Decryption Mode	00 : ECB	01 : OFB
8		10 : CBC	11 : (reserved)
9	Encryption Initialize	Not yet	Completed
10	Decryption Initialize	Not yet	Completed
11	Detect Active Tamper (All Keys erased)	Not Detect	Detect
12	Encryption Processing	Not busy	Busy
13	Decryption Processing	Not busy	Busy
14	Detect Active Tamper (RAM Keys erased)	Not Detect	Detect
15	ESN Timeout	Not Timeout	Timeout

Fig. 4-20 Cipher Status Data Format

4.4.11 Firmware Update

Update the module's firmware. Cease all cryptographic operation, and transit to a mode communicating with the PC. The module will calculate the transferred firmware's CMAC value, and if the value equals with the one added with the firmware, the return ACK_FIRM_UPDT with its checksum, and perform update. If the CMAC value differs, the module will return ACK_FIRM_UPDT without checksum, and will not perform the update.

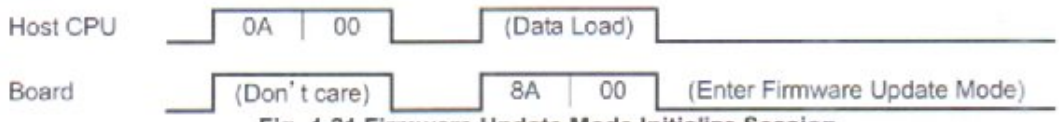


Fig. 4-21 Firmware Update Mode Initialize Session

Fig. 4-22 Firmware Update Mode Success Session (MAC equaled)

Fig. 4-23 Firmware Update Mode Failure Session (MAC did not equal.)

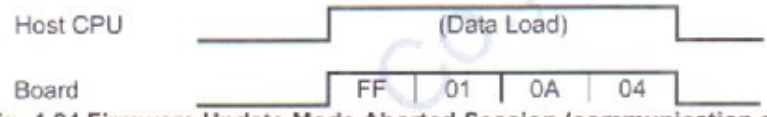


Fig. 4-24 Firmware Update Mode Aborted Session (communication error)

4.4.12 ESN Request

Request the ESN (16 word) number in the board.

The Board does not have ESN. Even though, the Board must correspond a data to the Host CPU.

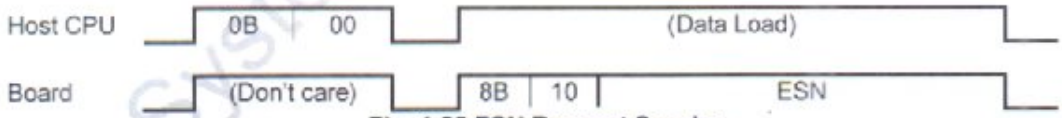


Fig. 4-25 ESN Request Session

4.4.13 Flash Memory ESN Request

Request the Flash Memory's ESN (8 word) written by the manufacturer (EON). All EON Flash Memory returns FFFF.

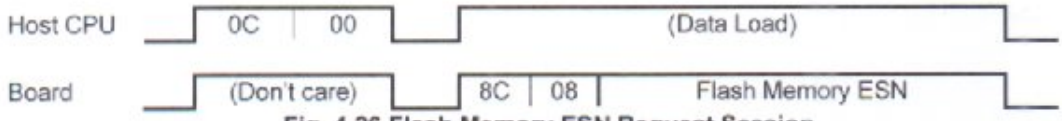


Fig. 4-26 Flash Memory ESN Request Session

4.4.14 Sleep Mode

Save the module's power consumption. The module shall be woken up for any operation. After sending ACK_SLEEP, PLL will be reset and powered down. In order to recover this mode, an interrupt request (/WAKEUP or Active Tamper detect) shall be made. If /WAKEUP is detected, ACK_IDLE is output and the module exits the sleep mode (discussed later).

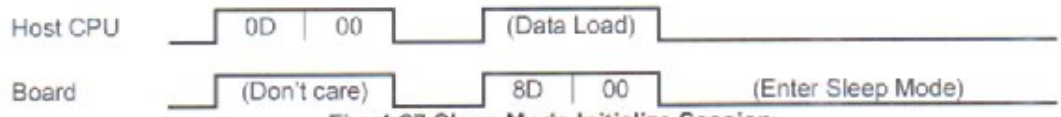


Fig. 4-27 Sleep Mode Initialize Session

4.4.15 Modify clock rate.

Modify the DSP's clock rate. After modification, ACK_CHG_CLK will be returned.

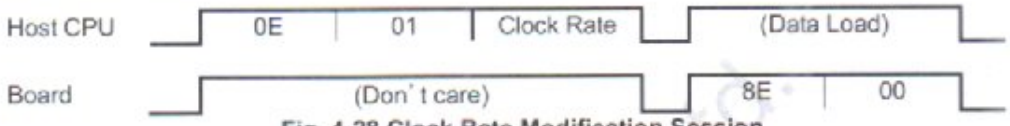


Fig. 4-28 Clock Rate Modification Session

Clock Rate

Code	Clock Rate
0000	1/2
0001	1
0002	2
0003	3
0004	4
0005	5
0006	6
0007	7
0008	8
0009	9
000A	10

Fig. 4-29 Clock Rate Configuration Value

4.4.16 Beat Shift

Shift the frequency of the clock generator. As the beat shift's control end (output Lo or Hi at HD2), ACK_BSHIFT is returned. The 2 word is 0000 when Beat Shift off, and 0001 when Beat Shift on.

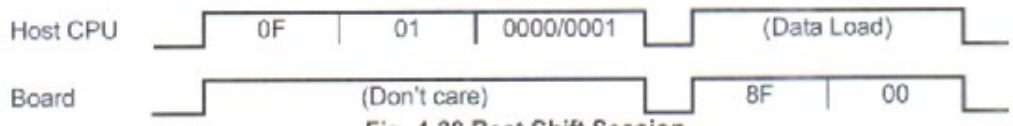


Fig. 4-30 Beat Shift Session

4.4.17 Version/Checksum Request

Request the DSP's firmware version number and checksum. A version number (ASCII code) and the checksum is returned.

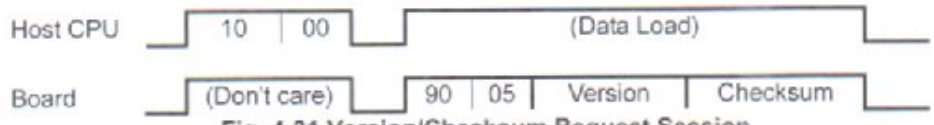


Fig. 4-31 Version/Checksum Request Session

Version

WORD	Upper 8 bits	Lower 8 bits
1	'A' or 'D' 'A' : KWD-AE30/AE31 'D' : KWD-DE31	'3' or '4' '3' : KWD-AE30 '4' : KWD-AE31/DE31
2	'.' (period)	'*' (Sub version)
3	'.' (period)	'*' (Minor version)
4	' ' (space)	' ' (space)

4.4.18 Invalid Response

Note that the command is invalid to the Host CPU. Invalid commands and/or invalid access will result in this response. The upper 8 bits is the command number, followed by the reason code in the lower 8 bits.

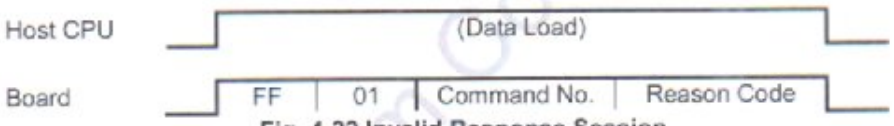


Fig. 4-32 Invalid Response Session

Reason Code

Code	Reason
xx00	(reserved)
xx01	Invalid 1st Command
xx02	Invalid 2nd Command
xx03	Invalid Data Length
xx04	Data Receive Error
xx05	Invalid CKR
xx06	Invalid KID
xx07	Invalid ALGID
xx08	Invalid Encryption/Decryption Mode
xx09	Encryption/Decryption Initialize Error
xx0A	Not Initialized
xx0B	Encryption/Decryption Error
xx0C	Encryption/Decryption was Busy
xx0D	Key was Not Found
xx0E	RNG Error
xx0F	Invalid SID

Fig. 4-33 Reason Code

4.4.19 Transition to Idle State

After reset and the module are initialized successfully, this command is output, and the module transits to an Idle state. Also, if a /WAKEUP interrupt occurred during sleep mode, this command is output and the module transits to an Idle state.

Fig. 4-34 Transition to Idle State Session

4.4.20 SKL Mode (not supported for KWD-AE30)

Session to communicate with Software Key Loader (KPG-113AE/114DE/151AE. Here in after, SKL)

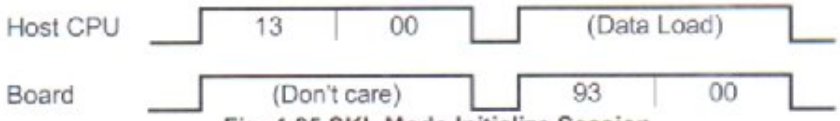


Fig. 4-35 SKL Mode Initialize Session

- When terminating the session, send data in the below format.
  Status's data format is same with KVL mode's (Fig4.17).
- When the SKL mode terminates, Message ID = 0x0045 / Status = 0x0000 will be sent.

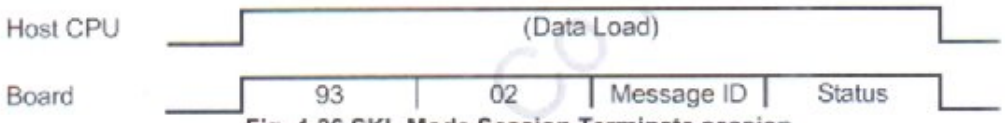


Fig. 4-36 SKL Mode Session Terminate session

4.4.21 Model Name Request

Host CPU requests Board's Model Name. Area will be reserved that may be utilized to show update functions in the future.

The Board should correspond a fixed value for this command.

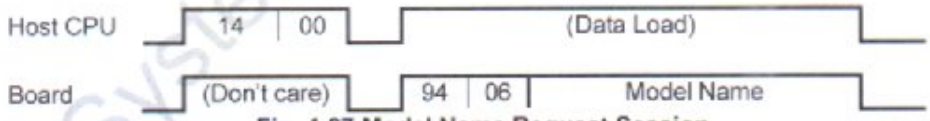


Fig. 4-37 Model Name Request Session

Model Name

WORD	Upper 8 bit	Lower 8 bit
1	Model Type 'S' : Means SCM	Model Number 1st byte KWD-AE3* : 'A' KWD-DE3* : 'D'
2	Model Number 2nd byte 'E'	Model Number 3rd byte '3'
3	Model Number 4th byte '0' or '1'	Type (upper 4 bits) K : 0000b Frequency version (lower 4 bits Unused : 1111b
4	Option (reserved) (0xFFFF in KWD-AE30/AE31/DE31)	
5	Maximum Key storage number (0x0400 in KWD-AE30/AE31/DE31)	
6	Maximum CKR Number (0xFFFF in KWD-AE30/AE31/DE31)	

4.4.22 AM Calculate

Calculate Authentication Mechanism (AM) processed under Link Layer Authentication. Expand a 80 bit RS to 128bit, and if the mode is AM3-AM4, reverse the bits and correspond to the specified SUID. Use a 128 bit encryption key to perform AES-128 (ECB mode) to get a 128 bit KS encryption key. Expand a 40 bit RAND to 128 bit, and use the 128 bit KS encryption key to perform AES (ECB) mode, and return a compressed 32 bit RS.

Fig. 4-38 AM Calculate session

SUID : WACN ID, System ID, Subscribe ID (4 WORD) Refer to the following table for ID
RS : Random Seed (5 WORD)
RAND : Random Challenge (3 WORD)
Mode : AM1-AM2 (0000), AM3-AM4 (0001)
RES : Response (2 WORD)

SUID

WORD	Superior 8 bit	Lower 8 bit
1	WACN ID (19-12)	WACN ID (11-4)
2	WACN ID (3-0) / System ID (11-8)	System ID (7-0)
3	Subscriber ID (23-16)	Subscriber ID (15-8)
4	Subscriber ID (7-0)	All 0

- If an invalid SUID is specified, INVALID_RSP is returned.

4.4.23 SUID Report

Return the SUID (WACN ID, System ID, Subscriber ID) of the radio to SCM. When using a module that supports Link Layer Authentication, this command shall be used to report the radio's SUID to SCM before communicating with the Key Loader (KVL-4000) using KVL Mode On command. The returned SUID will use 4 WORD upon each SUID, and a single command may return up to 32 SUID. The radio will return all SUID, up to 64, and will also report if the SUID is active or not.

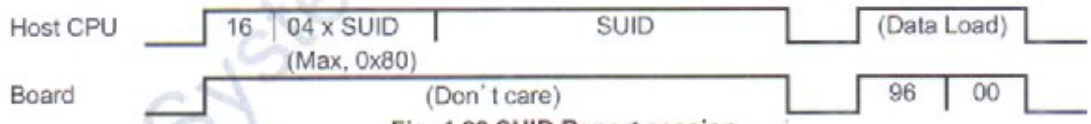


Fig. 4-39 SUID Report session

The returned SUID (4WORD) is as follow.

SUID

WORD	Superior 8 bit	Lower 8 bit
1	WACN ID (19-12)	WACN ID (11-4)
2	WACN ID (3-0) / System ID (11-8)	System ID (7-0)
3	Subscriber ID (23-16)	Subscriber ID (15-8)
4	Subscriber ID (7-0)	Active : 01 / Inactive : 00

4.4.24 Encryption Initialization MSV (Multi-Stream Version) (not supported for KWD-AE30)

Encryption initialization process. Any encryption shall take place after this session. Key expansion, CRNG tests are performed, and after initialization, Algorithm ID, Key ID, IV (MI) and SID is returned according to the CKR. Encryption can be initialized by either specifying the CKR or specifying both the Key ID and Algorithm ID.

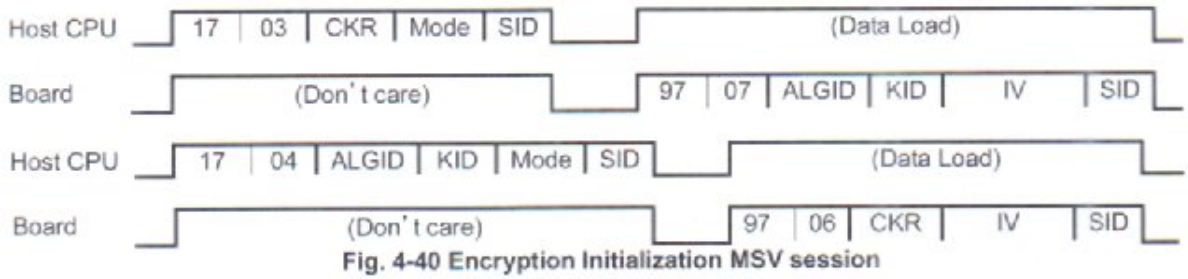


Fig. 4-40 Encryption Initialization MSV session

CKR : 0001 ~ FFFF
Mode : OFB Mode (0000), ECB Mode (0001)
ALGID : DES (0081), AES-256 (0084), AES-128 (0085)
KID : 0000 ~ FFFF
SID : Stream ID (0x0000 ~ 0x000F)
IV : Any 4 word except for all 0

- If an invalid mode and/or algorithm ID, SID is assigned, an INVALID_RSP is returned.
- If an invalid CKR (i.e., there are no such keys with the CKR) is assigned, an INVALID_RSP is returned.
- If the CRNG test fails, an INVALID_RSP is returned, and the module transits to an error state.

4.4.25 Decryption Initialization MSV (Multi-Stream Version) (not supported for KWD-AE30)

Decryption initialization process. Any decryption shall take place after this session. The host will send information with the following order; Algorithm ID, Key ID, Mode, IV(MI) and SID. After initialization including key expansion, return the CKR and SID.

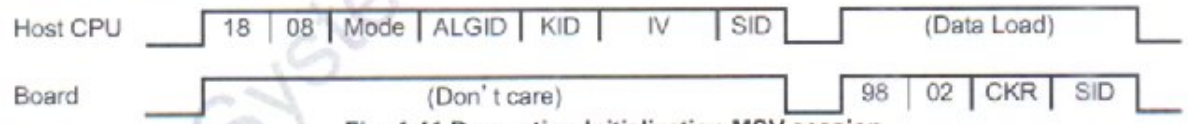


Fig. 4-41 Decryption Initialization MSV session

CKR : 0001 ~ FFFF
Mode : OFB Mode (0000), ECB Mode (0001)
ALGID : DES (0081), AES-256 (0084), AES-128 (0085)
KID : 0001 ~ FFFF
IV : Any 4 word except for all 0
SID : Stream ID (0x0000 ~ 0x000F)

- If an invalid mode and/or algorithm ID, SID is assigned, and INVALID_RSP is returned.
- If an invalid algorithm ID / Key ID (i.e., there are no such keys with the algorithm ID / Key ID) pair is assigned, an INVALID_RSP is returned.

4.4.26 Encryption MSV (Multi-Stream Version) (not supported for KWD-AE30)

Encryption. A ciphertext is output with the given plaintext. Initialization shall take place before this function. After the command, the Host CPU adds the plaintext required to encrypt. DES is 4 Word and AES is 8 Word. And then 1 Word of SID is followed. After encryption, the ciphertext is returned with SID. If initialization is not performed, an INVALID_RSP is returned.

Fig. 4-42 Encryption MSV session

SID : Stream ID (0x0000 ~ 0x000F)

- If an invalid SID is assigned, an INVALID_RSP is returned.

4.4.27 Decryption MSV (Multi-Stream Version) (not supported for KWD-AE30)

Decryption. A plaintext is output with the given ciphertext. Initialization shall take place before this function. After the command, the Host CPU adds the ciphertext required to decrypt. DES is 4 Word and AES is 8 Word. And then 1 Word of SID is followed. After decryption, the plaintext is returned with SID. If initialization is not performed, an INVALID_RSP is returned.

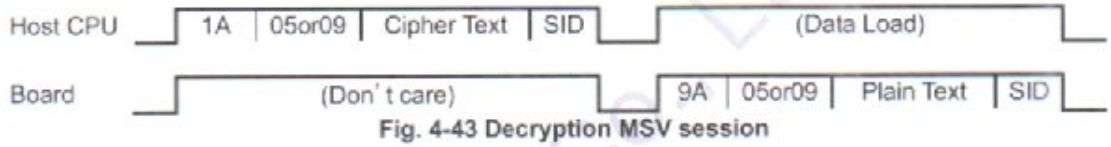


Fig. 4-43 Decryption MSV session

SID : Stream ID (0x0000 ~ 0x000F)

- If an invalid SID is assigned, an INVALID_RSP is returned.

4.4.28 Status Request MSV (Multi-Stream Version) (not supported for KWD-AE30)

The Host CPU can interrogate the SID module status to the Board. If the Board is in error condition,it should correspond this Status Request command only. The Self Test Status is also using at the Power Up Selt Test session.

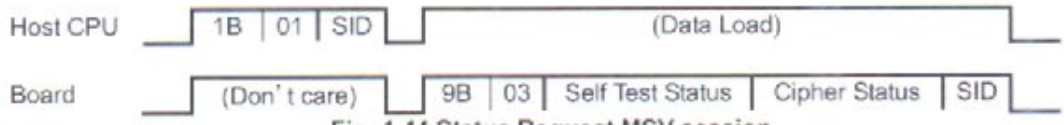


Fig. 4-44 Status Request MSV session

Self Test Status

Bit	Function	0	1
0	Power Up Self Test	Completed	Testing
1	Firmware Integrity Test	Success	Error
2	DES Encryption KAT	Success	Error
3	AES Encryption KAT	Success	Error
4	DES Decryption KAT	Success	Error
5	AES Decryption KAT	Success	Error
6	LFSR KAT	Success	Error
7	SHA-256 KAT	Success	Error
8	CMAC KAT	Success	Error
9	Firmware Update	Success	Error
10	Continuous RNG Test	Completed	Testing
11	Continuous RNG Test Result	Success	Error
12	(reserved)	Always 0	
13	(reserved)	Always 0	
14	(reserved)	Always 0	
15	(reserved)	Always 0	

Fig. 4-45 Self Test Status data format

Cipher Status

Bit	Function	0	1
0	FIPS Mode	Not validated mode	Validated mode
1	(reserved)	Always 0	
2	Encryption Algorithm	DES	AES
3	Decryption Algorithm	DES	AES
4	(reserved)	Always 0	
5	Encryption Mode	00 : ECB	01 : OFB
6		10 : CBC	11 : (reserved)
7	Decryption Mode	00 : ECB	01 : OFB
8		10 : CBC	11 : (reserved)
9	Encryption Initialize	Not yet	Completed
10	Decryption Initialize	Not yet	Completed
11	Detect Active Tamper (All Keys erased)	Not Detect	Detect
12	Encryption Processing	Not busy	Busy
13	Decryption Processing	Not busy	Busy
14	Detect Active Tamper (RAM Keys erased)	Not Detect	Detect
15	ESN Timeout	Not Timeout	Timeout

Fig. 4-46 Cipher Status data format

SID : Stream ID (0x0000 ~ 0x000F)

If an invalid SID is assigned, an INVALID_RSP is returned

4.4.29 OTAR KMM Command

Requests command process for OTAR KMMs. KMM's data size will differ respectively with the message, so the word number will change each time. When key modification/deletion is performed, encryption/decryption will require initialization.

Fig. 4-47 OTAR KMM Command Session

- Format of the command sent in the 1st word.

Upper 4 bits	Lower 12 bits
KMM Command 0110:Host CPU→SCM 1110:SCM→Host CPU	KMM word number 0x001-0xFFF(1-4095)

- Format of the KMM command sent from the 2nd word is as follows.

■ Calc-MAC-Req (HOST CPU→SCM)

Requests calculation of Checksum or Message Authentication Code (MAC) for a given message. Dependent of the type of the MAC, 3 formats is supported.

Type=0x01(Checksum)

01	Null	01	Null	Null	Null	Length	Message Data

Type=0x02(Enhanced)

01	Null	02	AlgID	KeyID	Fmt	Null	Length	Message Data

Type=0x03(Type3)

01	Null	03	AlgID	KeyID	Null	Length	Message Data

Fig. 4-48 Calc-MAC-Req Command

AlgID : Algorithm ID(0x81=DES,0x84=AES)
KeyID : Key ID Fmt : MAC Format(0x40=CBC MAC,0x41=CMAC)
Length : Length of Message Data[byte] If the byte size is an odd number, Message data will be left aligned.

■ Calc-MAC-Cnf (SCM→HOST CPU)

Response for the request to calculate Checksum or MAC. There are three types of responses, dependent on the given condition.

1. If the Checksum or MAC is successfully generated

81	00	Type	Null	Data

For Type=02(Enhanced), insert MAC Format to the Null area

2. If the MAC key did not exist (item does not exist)

81	01

3. If an invalid parameter was specified (not performed)

81	04

Data : Calculated Checksum or MAC (Type1 or 3=4bytes Type2=8bytes)

Fig. 4-49 Calc-MAC-Cnf Command

■ Set-Key-Req (HOST CPU→SCM)

Requests to store encryption keys. All encryption keys will be wrapped with the KEK, so using the KEK stored in the SCM beforehand, the key will be unwrapped and then stored in RAM (plaintext) and ROM (ciphertext, using the ESN to derive the encryption key). When the specified Key Length is "zero", the key specified with the KeysetID and SLN will be deleted.

1-1.Set Key (Flash ROM access)

02	Null	KSID	KLen	SLN	Fmt	AlgID	KeyID	TMP	EAlg	KEKID	EncKey	KeyName

1-2.Set Key (RAM access only)

02	FF	KSID	KLen	SLN	Fmt	AlgID	KeyID	TMP	EAlg	KEKID	EncKey	KeyName

2. Delete Key

02	Null	KSID	KLen	SLN

Fig. 4-50 Set-Key-Req Command

KSID : KeysetID(0x01-0xFF)
KLen : Key Length(0x08=DES 0x28=AES 0x00=Delete Key)
SLN : Storage Location Number(0x0001-0xFFFF)
Fmt : Key Format(MSB=0:TEK MSB=1:KEK LSB5bit:Key Name Size 0-31bytes)
AlgID : Algorithm ID for Wrapped Key(0x81=DES,0x84=AES)
KeyID : Key ID for Wrapped Key
TMP : Temporary Key Indicator(0=False 1=True)
EAlg : Algorithm ID for Key Wrapping(0x81=DES,0x84=AES)
KEKID : Key ID for Key Wrapping
EncKey : Wrapped Key Data(DES:8bytes AES:40bytes)
KeyName : Key Name for Wrapped Key

■ Set-Key-Cnf(SCM→HOST CPU)

Response for the request to store the encryption key. There are 5 types of responses, dependent on the given condition.

1. If successfully stored (success)

82	00

2. If key to delete did not exist (item does not exist)

82	01

3. If memory to store the data ran out (out of memory)

82	02

4. If the KEK to unwrap the key did not exist (unable to decrypt)

82	03

5. If an invalid parameter was specified (not performed)

82	04

Fig. 4-51 Set-Key-Cnf Command

■ Set-KeysetInfo-Req (HOST CPU→SCM)

Requests to store Keyset information.

03	Null	KSID	AlgID	Fmt	KsetName	KsetName

Fig. 4-52 Set-KeysetInfo-Req Command

KSID : KeysetID(0x01-0xFF)
AlgID : Algorithm ID(0x81=DES,0x84=AES)
Fmt : Keyset Format(MSB=0:TEK MSB=1:KEK LSB5bit:Keyset Name Size 0-31bytes)
KeyName : Keyset Name

■ Set-KeysetInfo-Cnf (SCM→HOST CPU)

Response for the request to store Keyset information. There are three types of responses, dependent on the given condition.

1. If performed successfully (success)

83	00

2. If the specified Keyset did not exist (item does not exist)

83	01

3. If an invalid parameter was specified (not performed)

83	04

Fig. 4-53 Set-KeysetInfo-Cnf Command

■ Set-RSI-Req (HOST CPU→SCM)

Requests to store RSI information.

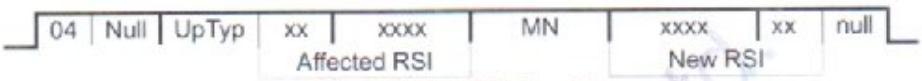


Fig. 4-54 Set-RSI-Req Command

UpTyp : Update Type(0x00=Add 0x01=Delete 0x02=Change 0x03=Set Message Number)
MN : Message Number for add RSI / Change RSI / Set Message Number
Affected RSI : RSI to be added, deleted, changed, or to have its Message Number set
New RSI : New RSI value for changed RSI

■ Set-RSI-Cnf (SCM→HOST CPU)

Response for the request to store RSI information. There are four types of responses, dependent on the given condition.

1. If performed successfully (success)

84	00

2. If the specified RSI did not exist (item does not exist)

84	01

3. If two or more arbitrary Group RSIs were intended to store (out of memory)

84	02

4. If an invalid parameter was specified (not performed)

84	04

Fig. 4-55 Set-RSI-Cnf Command

■ Delete-Key-Req (HOST CPU→SCM)

Requests to delete the specified key

1. Flash ROM access

05	Null	AlgID	Null	KeyID

2. RAM access only

05	FF	AlgID	Null	KeyID

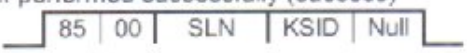
Fig. 4-56 Delete-Key-Req Command

AlgID : Algorithm ID(0x81=DES,0x84=AES)
KeyID : Key ID

■ Delete-Key-Cnf (SCM→HOST CPU)

Response for the request to delete the key. There are two types of responses, dependent on the given condition.

1. If performed successfully (success)



SLN : Storage Location Number of Deleted Key
KSID : KeysetID of Deleted Key

2. If the specified key did not exist (item does not exist)

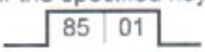


Fig. 4-57 Delete-Key-Cnf Command

■ Delete-Keyset-Req (HOST CPU→SCM)

Requests to delete Keyset information.

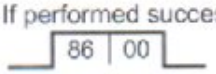
Fig. 4-58 Delete-Keyset-Req Command

KSID : KeysetID(0x01-0xFF)

■ Delete-Keyset-Cnf (SCM→HOST CPU)

Response for the request to delete Keyset information. There are two types of responses, dependent on the given condition.

1. If performed successfully (success)



2. If the specified Keyset did not exist (item does not exist)

Fig. 4-59 Delete-Keyset-Cnf Command

■ Changeover-Keyset-Req (HOST CPU→SCM)

Requests to change the Active Keyset.

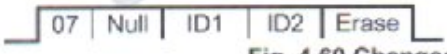


Fig. 4-60 Changeover-Keyset-Req Command

ID1 : ID of the Keyset that needs to be superseeed (0x01-0xFF)
ID2 : ID of the Keyset that needs to be activated (0x01-0xFF)
Erase : Erase frag for previous Keyset(0=False 1=True)

■ Changeover -Keyset-Cnf (SCM→HOST CPU)

Response for the request to change the Active Keyset. There are three types of responses, dependent on the given condition.

1. If performed successfully (success)

87	00	ID1	ID2

2. If the specified Keyset did not exist (item does not exist)

87	01	ID1	ID2

3. If an invalid parameter was specified (not performed)

87	04	ID1	ID2

Fig. 4-61 Changeover -Keyset-Cnf Command

■ Zeroize-Req (HOST CPU→SCM)

Requests to delete all keys.

08	Null

Fig. 4-62 Zeroize-Req Command

■ Zeroize-Cnf (SCM→HOST CPU)

Response for the request to zeroize all keys.

88	00

Fig. 4-63 Zeroize-Cnf Command

■ Get-KeyIDs-Req (HOST CPU→SCM)

Requests to output all key information stored.

09	Null

Fig. 4-64 Get-KeyIDs-Req Command

■ Get-KeyIDs-Cnf (SCM→HOST CPU)

Response for the request to output all key information stored. There are two types of responses, dependent on the given condition.

1. If performed successfully (success)

89	00	Key count	SLN	AlgID	Null	KeyID	···

Key count : Number of KeyIDs in storage
For the number of stored keys specified in the "Key count", SLN, AlgID, KeyID will be repeated.
SLN : Storage Location Number(0x0001-0xFFFF)
AlgID : Algorithm ID (0x81=DES,0x84=AES)
KeyID : Key ID

2. If key did not exist (item does not exist)

89	01

Fig. 4-65 Get-KeyIDs-Cnf Command

■ Get-KeysetIDs-Req (HOST CPU→SCM)

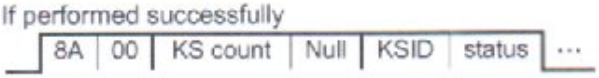
Requests to output all Keyset information stored.

Fig. 4-66 Get-KeysetIDs-Req Command

■ Get-KeysetIDs-Cnf (SCM→HOST CPU)

Response for the request to output all Keyset information stored. There are two types of responses, dependent on the given condition.

1. If performed successfully



KS count : Number of KeysetIDs in storage
For the number of stored Keysets specified in the "KS count", KeysetID, status will be repeated.
KSID : KeysetID(0x01-0xFF)
status : 0x00=Inactive 0x01=Active

2. If Keyset did not exist (item does not exist)

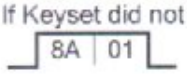


Fig. 4-67 Get-KeysetIDs-Cnf Command

■ Get-KeysetKeyIDs-Req (HOST CPU→SCM)

Requests to output the key information within the specified Keyset.

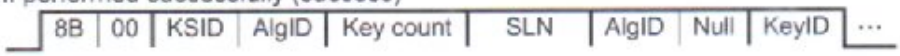
KSID : KeysetID(0x01-0xFF)

Fig. 4-68 Get-KeysetKeyIDs-Req Command

■Get-KeysetKeyIDs-Cnf(SCM→HOST CPU)

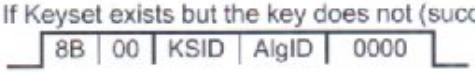
Response for the request to output key information within the specified Keyset. There are two types of responses, dependent on the given condition.

1. If performed successfully (success)



KSID : KeysetID(0x01-0xFF)
AlgID : Algorithm ID (0x81=DES,0x84=AES)
Key count : Number of KeyIDs
For the number of stored key stored within the specified Keysets noted in the "Key count", SLN, AlgID, KeyID will be repeated.
SLN : Storage Location Number(0x0001-0xFFFF)
KeyID : Key ID

2. If Keyset exists but the key does not (success)



3. If Keyset does not exist (item does not exist)

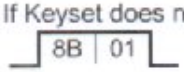


Fig. 4-69 Get-KeysetKeyIDs-Cnf Command

■ Get-KeysetInfo-Req (HOST CPU→SCM)

Requests for the specified Keyset information.

KSID : KeysetID(0x01-0xFF)

Fig. 4-70 Get-KeysetInfo-Req Command

■ Get-KeysetInfo-Cnf (SCM→HOST CPU)

Response for the request to output the specified Keyset. There are two types of responses, dependent on the given condition.

1. If performed successfully (success)

AlgID : Algorithm ID(0x81=DES,0x84=AES)
Fmt : Keyset Format(MSB=0:TEK MSB=1:KEK LSB5bit:Keyset Name Size 0-31bytes)
KeyName : Keyset Name

2. If the specified Keyset did not exist (item does not exist)

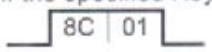


Fig. 4-71 Get-KeysetInfo-Cnf Command

■ Get-RSI-Req (HOST CPU→SCM)

Requests to output RSI information.

RSI : 0x00000000-0x0098967F : Individual RSI
0x00989680-0x00FFFFFF : Group RSI
0x01000000 : Individual and KMF RSI
0x01000001 : All known RSIs

Fig. 4-72 Get-RSI-Req Command

■ Get-RSI-Cnf (SCM→HOST CPU)

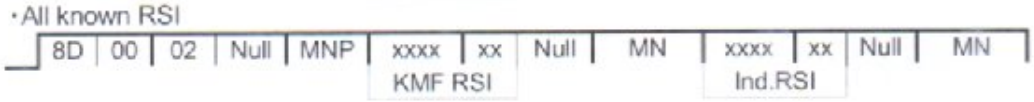
Response for the request to output RSI Information. There are four types of responses, dependent on the given condition.

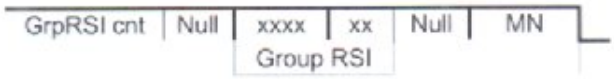
1. If performed successfully (success)

·Single RSI

·Individula and KMF RSI

·All known RSI





MNP : Message Number Period
RSI : Radio Set ID
MN : Message Number
Ind.RSI : Individual RSI
Grp.RSI cnt : Group RSI count (0 or 1)
Group RSI : Group RSI (0x000000="none")

2. If the specified RSI did not exist (item does not exist)

Fig. 4-73 Get-RSI-Cnf Command

■ Get-Capabilities-Req (HOST CPU→SCM)

Requests to output Option Service IDs and Message IDs supported

Fig. 4-74 Get-Capabilities-Req Command

■ Get-Capabilities-Cnf (SCM→HOST CPU)

Response for the request to output Options Service IDs and Message IDs supported.

Fig. 4-75 Get-Capabilities-Req Command

Cnt : Algorithm ID count (AES&DES=0x02)
AlgID : Algorithm ID (0x81=DES,0x84=AES)
Op.cnt : Option Service ID count (0x05)
Op.Svs.IDs : Option Service IDs
- 0x06 "2octet Message Numbers"
- 0x09 "Checksum"
- 0x0B "Type3/Type4 MAC"
- 0x0C "Key Name is supported"
- 0x0D "Keyset Name is supported"

MID.cnt : Message ID count (0x18)
Msg.IDs : Message ID
- 0x01 "Capabilities-Command"
- 0x02 "Capabilities-Response"
- 0x03 "Change-RSI-Command"
- 0x04 "Change-RSI-Response"
- 0x05 "Changeover-Command"
- 0x06 "Changeover-Response"
- 0x07 "Delayed-Acknowledgement"
- 0x08 "Delete-Key-Command"
- 0x09 "Delete-Key-Response"
- 0x0A "Delete-Keyset-Command"
- 0x0B "Delete-Keyset-Response"
- 0x0C "Hello"
- 0x0D "Inventory-Command"
- 0x0E "Inventory-Response"
- 0x13 "Modify-Key-Command"
- 0x14 "Modify-Keyset-Attributes-Command"
- 0x15 "Modify-Keyset-Attributes-Response"
- 0x16 "Negative-Acknowledgement"
- 0x17 "No-Service"
- 0x1D "Rekey-Acknowledgement"
- 0x1E "Rekey-Command"
- 0x20 "Warm-Start-Command"
- 0x21 "Zeroize-Command"
- 0x22 "Zeroize-Response"

5. Key Loader

The Key programming procedure is folloiwng the EIA/TIA-102 AACD.

http://global.ihs.com/doc_detail.cfm?currency_code=USD&customer_id=2125452A320A&oshid=2125452A3A0A&shopping_cart_id=2125452A3B0A&rid=TIA&input_doc_number=TIA-102&country_code=US&lang_code=ENGL&item_s_key=00457443&item_key_date=880831&input_doc_number=AACD&input_doc_title=&org_code=TIA

5.1 Sequense

There is an example when connect/disconnect the KVL-3000 to the radio.

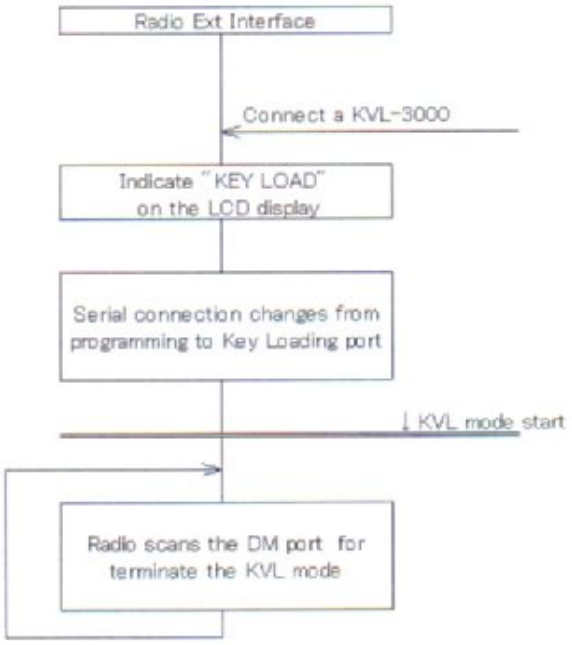


Fig. 5-1 Connect KVL-3000

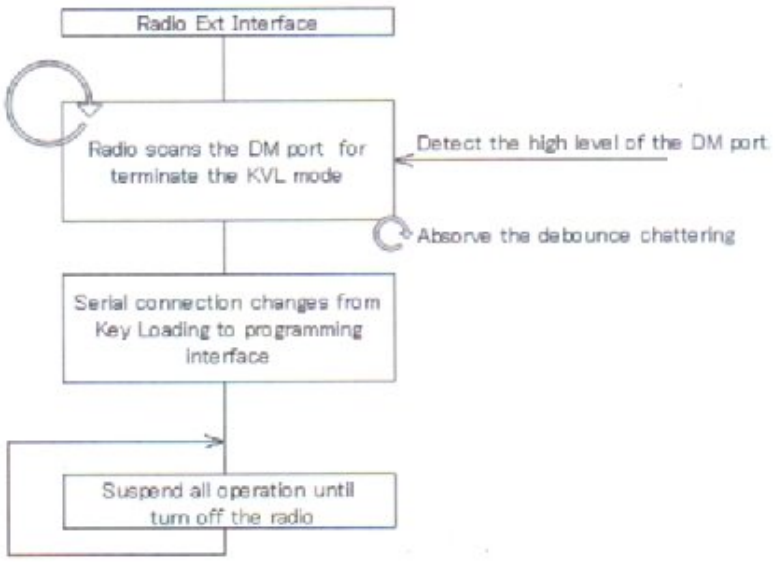


Fig. 5-2 Disconnect KVL-3000

6. Hardware

6.1 Terminal Function

The Board has to adapting the following specification.

Connector: 20pin Board to Board Connector

pin No.	pin name	I/O	function	H/L	impedance
1	GND	-	GND	-	-
2	GND	-	GND	-	-
3	/RESET	I	Reset	High:2.4V to Vcc+0.3V/Low:0.0 to 0.8V	>47kΩ
4	TXD	O	UART(KVL) data	High:2.4V to Vcc / Low:0.0 to 0.4V	<1kΩ
5	SCK	I	SPI shift clock	High:2.4V to Vcc+0.3V/Low:0.0 to 0.8V	>47kΩ
6	RXD	I	UART(KVL) data	High:2.4V to Vcc+0.3V/Low:0.0 to 0.8V	>47kΩ
7	/REQ	O	interrupt request	High:2.4V to Vcc / Low:0.0 to 0.4V	<1kΩ
8	BUSY	O	busy indicator	High:2.4V to Vcc / Low:0.0 to 0.4V	<1kΩ
9	TAMPER2	-	Tamper2	Connect to GND	-
10	NC	-	No Connect	-	-
11	TAMPER	I	Tamper	MIN:1.8 MAX:5.5V	-
12	Vcc	I	+3.3V	MIN:2.7V TYP:3.3V MAX:3.6V **	-
13	MOSI	I	SPI data	High:2.4V to Vcc+0.3V/Low:0.0 to 0.8V	>47kΩ
14	BCLK	I	clock for McBSP	High:2.4V to Vcc+0.3V/Low:0.0 to 0.8V	>47kΩ
15	/SS	I	SPI slave enable	High:2.4V to Vcc+0.3V/Low:0.0 to 0.8V	>47kΩ
16	MISO	O	SPI data	High:2.4V to Vcc / Low:0.0 to 0.4V	<1kΩ
17	/WAKEUP	I	wakeup from sleep mode	High:2.4V to Vcc+0.3V/Low:0.0 to 0.8V	>22kΩ
18	/BFS	I	framesync for McBSP	High:2.4V to Vcc+0.3V/Low:0.0 to 0.8V	>47kΩ
19	GND	-	GND	-	-
20	GND	-	GND	-	-

6.2 Connector

Manufacture Matsushita Electric, Ltd

http://www3.panasonic.biz/ac/e/control/connector/base-fpc/p4/size_figure/index.jsp

Model P4 series (0.4mm pitch)

6.3 Board Size (KWD-AE30/AE31)