



# AW140

USER'S MANUAL

## FIPS 140-2 High Security Encryption Module

*Industrial-grade, long-range wireless Ethernet systems*

 **AvaLAN**  
WIRELESS

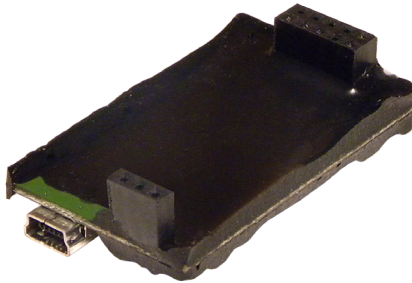
Thank you for your purchase of the AW140 FIPS 140-2 High Security Encryption Module.

This module from AvaLAN Wireless Systems is a multi-chip embedded security module that performs AES 128/192/256 bit encryption. It meets the FIPS 140-2 Level 2 Security Standard and is approved by CMVP (the Cryptographic Module Validation Program) a joint effort of NIST (The National Institute of Standards and Technology) and CESC (Communications Security Establishment Canada). This approval indicates acceptance by the Federal Agencies of both countries for the protection of sensitive information.

The module has a tamper-evident coating and two separate interfaces:

- A Serial Peripheral Interface (SPI) that is used to connect to a host microcontroller and transfers encrypted and non-encrypted data.
- A USB Interface that is used to establish login credentials, configure the encryption method and set the key.

If you have any questions when configuring your AvaLAN product, the best place to get answers is to visit [www.avalanwireless.com](http://www.avalanwireless.com). If more assistance is needed, send email to [support@avalanwireless.com](mailto:support@avalanwireless.com). To speak to a live technician, please call technical support at the number below during normal business hours.



© 2010 by AvaLAN Wireless Systems Inc. All rights reserved.

Revision 05.17.2010

125A Castle Drive  
Madison, AL 35758

Sales: (866) 533-6216  
Technical Support: (650) 384-0000  
Customer Service: (650) 641-3011  
Fax: (650) 249-3591

## Table of Contents

---

Features and Specifications . . . . .	4
Secure Setup and Initialization. . . . .	5
Module Physical Interface . . . . .	8
Module SPI Interface . . . . .	9
Appendix . . . . .	16

## Features and Specifications

- Government approved FIPS 140-2 Security
- Fast real time AES encryption at 5 Mbps
- Easily integrated SPI (Serial Peripheral Interface)
- Security key managed with a terminal interface via a USB port
- Tamper-evident conformal coating
- Small physical size
- Low power consumption
- Wide Temperature range: -40° to +85° C

Characteristic	Specification
Encryption Throughput	5 Mbps
Certifications	NIST FIPS 140-2, NIST FIPS 197
SPI Data Interface	2 x 5 pin header, 0.1" spacing plus 3 pin support
Command Interface	Mini USB Socket
Size	30 mm wide, 50 mm long, 15 mm deep including connector pins
Voltage Range	3.1 to 3.6 VDC
Power Consumption	45 ma at 3.3 VDC (150 mW)
Operating Temperature Range	-70° C to +80° C

## Secure Setup and Initialization

To program the encryption characteristics of the AW140 module, a PC must be connected to the USB port and the SPI interface must be powered off. The necessary portion of the electronics for managing the setup is powered from the USB connection.

The PC must have two critical pieces of software installed:

1. A driver that provides a virtual COM port through the USB connection. This driver can be downloaded from the Future Technology Devices International web site, <http://www.ftdichip.com>. Follow their menu to the web page for VCP drivers and choose the one that matches your operating system. Installation guides are also available in the documents section of the same web site.
2. A terminal emulator that will provide the user interface to the AW140. Windows XP and earlier included a program known as HyperTerminal that would work fine, but Microsoft has not included it in newer versions. There are many free or low cost alternatives available. A good choice for Windows systems as of this manual's release is RealTerm, available from <http://realterm.sourceforge.net>. For other operating systems, you will need to search the Internet for something suitable such as minicom for Linux Ubuntu or ZTerm for MAC.

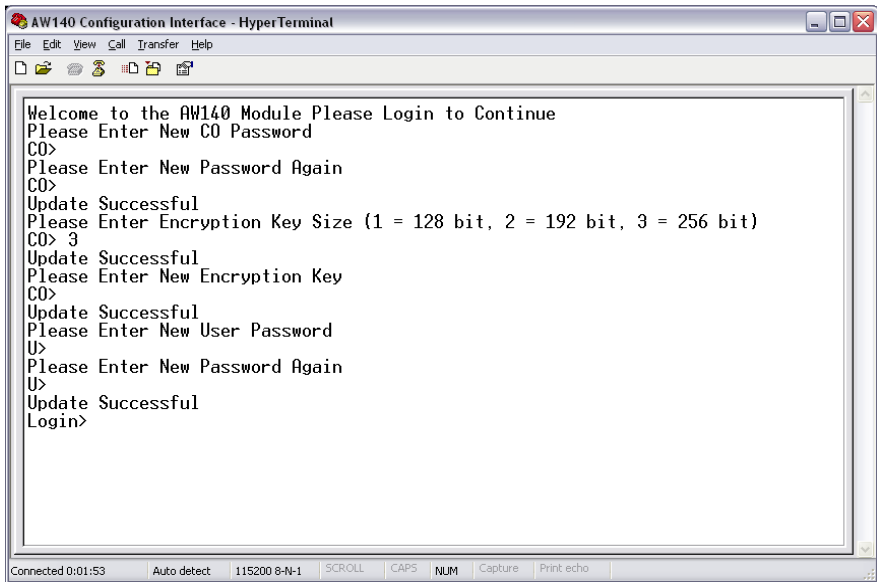
There are two roles defined for those having access to the programming interface, Crypto Officer and User. Each has a different password. Only the Crypto Officer is allowed to set the encryption method and encryption key. The User may examine self test results and firmware version only. The standard states that another responsibility of the Crypto Officer is to periodically examine the module for evidence of tampering.

Step by step programming procedure:

1. Connect the AW140 module's USB port to your computer using a USB mini B cable. If the module's SPI interface is wired into a microcontroller, the power for that circuitry should be off. The module can also be programmed stand alone, drawing power from the USB interface.
2. Open your terminal emulator program and set the COM port settings as follows:
  - Data bits: 8
  - Baud rate: 115200
  - Parity: none
  - Stop Bits: 1
  - Flow control: none

3. Press any key to activate the AW140. If the module has never been programmed, setup prompts will occur as shown in the example screen shot below. If you see only a login prompt, then the module has previously been initialized. If you know the password, enter it. If not, type "init" to erase all keys and passwords and return the module to its uninitialized state.

4. Initial Setup commands:



```
AW140 Configuration Interface - HyperTerminal
File Edit View Call Transfer Help
Welcome to the AW140 Module Please Login to Continue
Please Enter New CO Password
CO>
Please Enter New Password Again
CO>
Update Successful
Please Enter Encryption Key Size (1 = 128 bit, 2 = 192 bit, 3 = 256 bit)
CO> 3
Update Successful
Please Enter New Encryption Key
CO>
Update Successful
Please Enter New User Password
U>
Please Enter New Password Again
U>
Update Successful
Login>
```

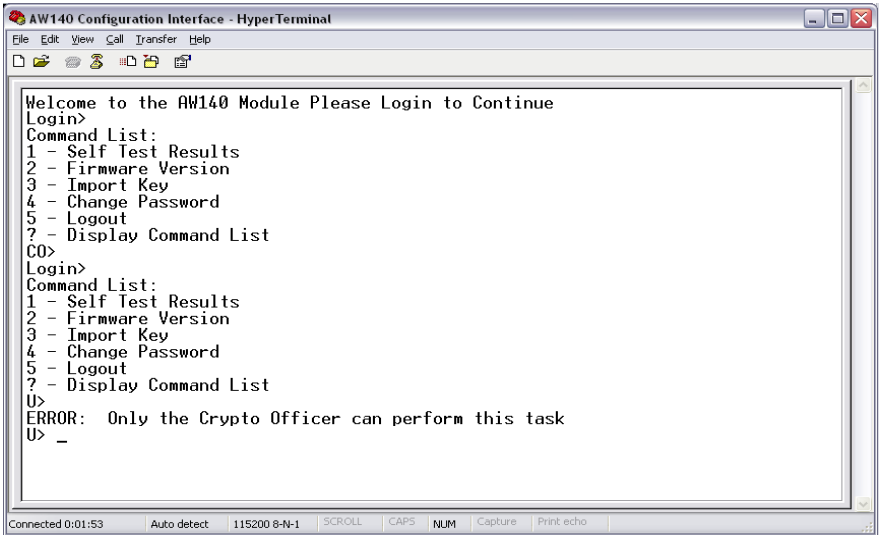
Passwords must be supplied for both the Crypto Officer and User Roles. Passwords must be between 8 and 32 characters. Case matters and any symbols may be used (all ASCII characters allowed).

The Encryption Key Size may be 128, 192, or 256 bits as shown in the prompt.

The Encryption Key must be entered as a 32, 48 or 64 digit hexadecimal number (0-9, a-f) corresponding to the Encryption Key Size. Here, case does not matter – A-F or a-f may be used. It is possible to enter less than the full number of digits; the AW140 will pad the rest with zeros.

5. After completing the initial setup, disconnect the USB cable and power up the AW140 via the SPI interface to begin normal cryptographic operation.

6. If it becomes necessary to change the programming or test the module at some later time, turn off the power for the SPI interface. Connect the USB cable to the AW140 and set up the COM port parameters and terminal emulator program as described in step 2. A screen similar to this one will appear:



```
AW140 Configuration Interface - HyperTerminal
File Edit View Call Transfer Help
Welcome to the AW140 Module Please Login to Continue
Login>
Command List:
1 - Self Test Results
2 - Firmware Version
3 - Import Key
4 - Change Password
5 - Logout
? - Display Command List
CO>
Login>
Command List:
1 - Self Test Results
2 - Firmware Version
3 - Import Key
4 - Change Password
5 - Logout
? - Display Command List
U>
ERROR: Only the Crypto Officer can perform this task
U> -
Connected 0:01:53 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Self Test Results displays the results of the power up self test. At power up, the AW140 runs a known answer test for all encryption/decryption algorithms.

Firmware Version displays the revision number of the firmware running in the AW140 module.

Import Key can only be used by the Crypto Officer Role. If the User Role attempts to run this command, an error occurs as shown in the screen shot.

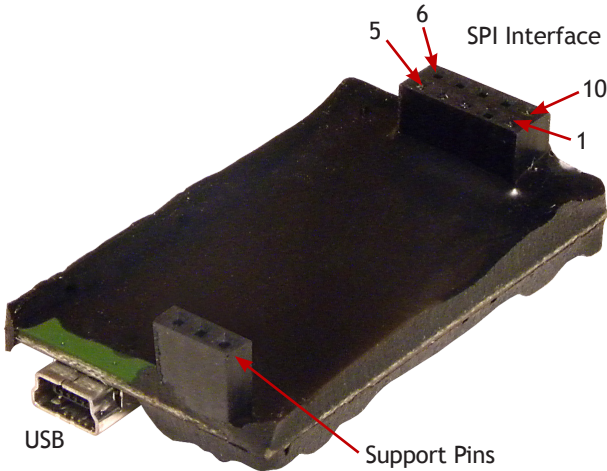
Change Password allows a new choice for the Crypto Officer or User password, depending on which Role is logged in.

Logout and Display Command List are self-explanatory.

Please note that if an incorrect password is entered at the login prompt, two more tries are allowed and then the AW140 enters a lockout state for 5 minutes.

7. After completing the setup or testing, log out and disconnect the USB cable. Then power up the AW140 via the SPI interface to resume normal cryptographic operation.

## Module Physical Interface



Here are the Signal definitions for the AW140 SPI Interface:

Pin Number	Name	Description
1	Vcc	3.3 vdc power for module
2	SCK	Serial Clock
3	MOSI	Serial Data Input to Module
4	MISO	Serial Data Output from Module
5	GND	Module Ground
6	#RESET	Active Low Reset
7	FIFO Full Flag	1=FIFO full, don't send any more data, 0=FIFO is empty
8	Data Ready	1=data packet available, 0=no data
9	Error	1=Error occured. To clear flag, deassert Chip Select
10	#CS	Active Low Chip Select



## Module SPI Interface

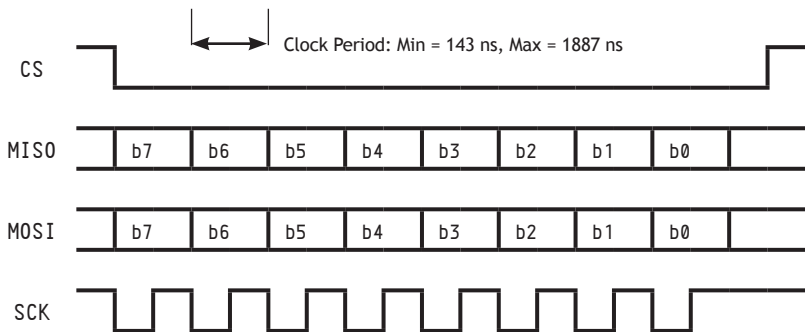
Serial Peripheral Interface (SPI) is a full duplex synchronous serial interface that allows data to be shifted in and out of the AW140 module 8 bits at a time, most significant bit first.

The SPI requires 4 pins to be physically connected:

- SCK - Serial bit shift clock (provided by master SPI)
- MISO - Master In Slave Out
- MOSI - Master Out Slave In
- CS - Active low Chip Select

The SPI on the AW140 Module is a slave SPI and uses mode (1,1) for clock phase and polarity. This means that the SCK line idles high and data is setup on the falling edge of the clock and latched on the rising edge.

The maximum clock rate for the SPI is 7MHz and the minimum clock rate is 530kHz.



The SPI operates in slave mode, meaning SCK is supplied by an external source. This interface is used to transfer data to and from the module and to read status information.

The first byte on the MOSI line after the #CS line goes low is the Command Byte. This byte tells the AW140 what command is to be executed.

### Command Byte:

b7	b6	b5	b4	b3	b2	b1	b0
get/set	-	-	-	-	-	CMD1	CMD0

When the get/set bit is set, information will be sent to the AW140 on MOSI and MISO

will be high impedance. When clear, a get transaction will take place and information will be sent from the AW140 on MISO.

After the command byte is issued the master microcontroller must delay to allow the AW140 enough time to prepare for the transaction. See the timing diagrams with each command for the delay times to use.

When a data transaction is complete and the #CS line is high, the master microcontroller must delay to allow the AW140 to finish processing the transaction.

CMD1 and CMD0 are used to tell the AW140 what command is to be executed according to this table:

Command Byte - HEX	Command
0x00	getStatus
0x01	getPlainText
0x02	getCipherText
0x03	getVersion
0x80	INVALID
0x81	setPlainText
0x82	setCipherText
0x83	setReset

## Status and Reset Commands

### getStatus:

The getStatus command is used to find out the current status of the module.

0x00	getStatus							
	b7	b6	b5	b4	b3	b2	b1	b0
Byte1	ST	DFIFO	EFIFO	DDATA	EDATA	CODE2	CODE1	CODE0

ST: When set, this bit indicates that the AW140 is performing power up self test.

DFIFO: When set, this bit indicates that the Decipher FIFO is full; no more ciphertext can be transferred to the module until some plaintext is read out.

EFIFO: When set, this bit indicates that the Encipher FIFO is full; no more plaintext can be transferred to the module until some ciphertext is read out.

DDATA: When set, this bit indicates that plaintext is ready to be read out of the

AW140.

EDATA: When set, this bit indicates that ciphertext is ready to be read out of the AW140.

CODE2...0: If an error occurs (Error line asserted), then there will be a condition code here. The error must be cleared by de-asserting the #CS line before operation can be resumed.

Error Code	Error
0x00	NULL/No code
0x01	Self Test in progress
0x02	Last command not understood
0x03	Data size invalid
0x04	No code
0x05	Self Test failed

#### getVersion:

The getVersion command is used to determine the firmware version running in the AW140 module.

0x03	getVersion							
	b7	b6	b5	b4	b3	b2	b1	b0
Byte1	Ma3	Ma2	Ma1	Ma0	Mi3	Mi2	Mi1	Mi0
Byte2	BN15	BN14	BN13	BN12	BN11	BN10	BN9	BN8
Byte3	BN7	BN6	BN5	BN4	BN3	BN2	BN1	BN0

Ma3...0: Major Version Number

Mi3...0: Minor Version Number

BN15...0: Build Number

#### setReset:

The setReset command is used to reset the AW140 and can be issued at any time during normal operation. After a reset has been issued the AW140 takes approximately 300ms to restart.

There are no other bytes required to reset the device. The host microcontroller simply needs to send the 0x83 Command Byte.

## Plain Data Commands

The Plain Data commands are used to transfer plaintext between AW140 and the host microcontroller.

### getPlainText:

The getPlainText command is used to read deciphered plaintext data from the AW140 module. The Data Ready line will be asserted and the EDATA bit of the Status register will be set when data is present in the decipher FIFO and will remain asserted until all data is read. There is protection for data in the decipher FIFO; the data will remain present until it has been read out.

0x01	getPlainText							
	b7	b6	b5	b4	b3	b2	b1	b0
Byte1	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

ID7...0: A packet identifier, the same one associated with the packet when it was sent to the AW140 using the setCipherText command.

	b7	b6	b5	b4	b3	b2	b1	b0
Byte2	-	-	-	-	-	S10	S9	S8
Byte3	S7	S6	S5	S4	S3	S2	S1	S0

S10...0: Data packet size in bytes.

	b7	b6	b5	b4	b3	b2	b1	b0
Byte4	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
...	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
ByteN	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

DATA7..0: Data bytes.

Timing requirements for getPlainText:

1. Between the Command Byte and Byte1, at least 4.0  $\mu$ s.
2. Between Byte1 and Byte2, Byte2 and Byte3, at least 0.5  $\mu$ s.
3. Between each data byte, at least 1.0  $\mu$ s.
4. After the last data byte and before de-asserting #CS, at least 2.0  $\mu$ s.

**setPlainText:**

The setPlainText command is used to submit data for encryption. The FIFO Full line will be asserted if the transmit FIFO cannot accept any more data. If the host microcontroller attempts to submit data while the FIFO Full line is asserted then the Error line will also become asserted and the data being submitted will not be entered into the FIFO.

Once the data has been fully transferred to the AW140 module, it is queued up for enciphering.

0x81	setPlainText							
	b7	b6	b5	b4	b3	b2	b1	b0
Byte1	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

ID7...0: A packet identifier, this value is associated with the data packet and will be sent back to the host microcontroller when the enciphered data is read back out using the getCipherText command.

	b7	b6	b5	b4	b3	b2	b1	b0
Byte2	-	-	-	-	-	S10	S9	S8
Byte3	S7	S6	S5	S4	S3	S2	S1	S0

S10...0: Data packet size in bytes. The number of bytes must be between 1 and 2047.

	b7	b6	b5	b4	b3	b2	b1	b0
Byte4	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
...	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
ByteN	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

DATA7..0: Data bytes.

Timing requirements for setPlainText:

- Between the Command Byte and Byte1, at least 1.8  $\mu$ s.
- Between Byte1 and Byte2, Byte2 and Byte3, at least 0.5  $\mu$ s.
- Between Byte3 and the first data byte, at least 2.0  $\mu$ s.
- Between each data byte, at least 1.1  $\mu$ s.
- After the last data byte and before de-asserting #CS, at least 4.5  $\mu$ s.

## Cipher Data Commands

The Cipher Data commands are used to transfer ciphertext between AW140 and the host microcontroller. They behave in a very similar manner to the Plain Data Commands just described.

### getCipherText:

The getCipherText command is used to read ciphered data from the AW140 module. The Data Ready line will be asserted and the DDATA bit of the Status register will be set when data is present in the encipher FIFO and will remain asserted until all data is read. There is protection for data in the encipher FIFO; the data will remain present until it has been read out.

0x02	getCipherText							
	b7	b6	b5	b4	b3	b2	b1	b0
Byte1	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

ID7...0: A packet identifier, the same one associated with the packet when it was sent to the AW140 using the setPlainText command.

	b7	b6	b5	b4	b3	b2	b1	b0
Byte2	-	-	-	-	-	S10	S9	S8
Byte3	S7	S6	S5	S4	S3	S2	S1	S0

S10...0: Data packet size in bytes.

	b7	b6	b5	b4	b3	b2	b1	b0
Byte4	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
...	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
ByteN	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

DATA7..0: Data bytes.

Timing requirements for getCipherText:

10. Between the Command Byte and Byte1, at least 4.0  $\mu$ s.
11. Between Byte1 and Byte2, Byte2 and Byte3, at least 0.5  $\mu$ s.
12. Between each data byte, at least 1.0  $\mu$ s.
13. After the last data byte and before de-asserting #CS, at least 2.0  $\mu$ s.

**setCipherText:**

The setCipherText command is used to submit data for decryption. The FIFO Full line will be asserted if the transmit FIFO cannot accept any more data. If the host microcontroller attempts to submit data while the FIFO Full line is asserted then the Error line will also become asserted and the data being submitted will not be entered into the FIFO.

Once the data has been fully transferred to the AW140 module, it is queued up for deciphering.

0x82	setCipherText							
	b7	b6	b5	b4	b3	b2	b1	b0
Byte1	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

ID7...0: A packet identifier, this value is associated with the data packet and will be sent back to the host microcontroller when the deciphered data is read back out using the getPlainText command.

	b7	b6	b5	b4	b3	b2	b1	b0
Byte2	-	-	-	-	-	S10	S9	S8
Byte3	S7	S6	S5	S4	S3	S2	S1	S0

S10...0: Data packet size in bytes. The number of bytes must be between 1 and 2047.

	b7	b6	b5	b4	b3	b2	b1	b0
Byte4	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
...	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
ByteN	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

DATA7..0: Data bytes.

Timing requirements for setCipherText:

14. Between the Command Byte and Byte1, at least 1.8  $\mu$ s.
15. Between Byte1 and Byte2, Byte2 and Byte3, at least 0.5  $\mu$ s.
16. Between Byte3 and the first data byte, at least 2.0  $\mu$ s.
17. Between each data byte, at least 1.1  $\mu$ s.
18. After the last data byte and before de-asserting #CS, at least 4.5  $\mu$ s.

