



Scientific Working Group on Digital Evidence

SWGDE Test Method for Bluetooth® Module Extraction and Analysis

Disclaimer:

As a condition to the use of this document and the information contained therein, the SWGDE requests notification by e-mail before or contemporaneous to the introduction of this document, or any portion thereof, as a marked exhibit offered for or moved into evidence in any judicial, administrative, legislative or adjudicatory hearing or other proceeding (including discovery proceedings) in the United States or any Foreign country. Such notification shall include: 1) the formal name of the proceeding, including docket number or similar identifier; 2) the name and location of the body conducting the hearing or proceeding; 3) subsequent to the use of this document in a formal proceeding please notify SWGDE as to its use and outcome; 4) the name, mailing address (if available) and contact information of the party offering or moving the document into evidence. Notifications should be sent to secretary@swgde.org.

It is the reader's responsibility to ensure they have the most current version of this document. It is recommended that previous versions be archived.

Redistribution Policy:

SWGDE grants permission for redistribution and use of all publicly posted documents created by SWGDE, provided that the following conditions are met:

1. Redistribution of documents or parts of documents must retain the SWGDE cover page containing the disclaimer.
2. Neither the name of SWGDE nor the names of contributors may be used to endorse or promote products derived from its documents.
3. Any reference or quote from a SWGDE document must include the version number (or create date) of the document and mention if the document is in a draft status.

Requests for Modification:

SWGDE encourages stakeholder participation in the preparation of documents. Suggestions for modifications are welcome and must be forwarded to the Secretary in writing at secretary@swgde.org. The following information is required as a part of the response:

- a) Submitter's name
- b) Affiliation (agency/organization)
- c) Address
- d) Telephone number and email address
- e) Document title and version number
- f) Change from (note document section number)
- g) Change to (provide suggested text where appropriate; comments not including suggested text will not be considered)
- h) Basis for change

Intellectual Property:

Unauthorized use of the SWGDE logo or documents without written permission from SWGDE is a violation of our intellectual property rights.

SWGDE Test Method for Bluetooth® Module Extraction and Analysis

Version: 1.1 (September 17, 2020)

This document includes a cover page with the SWGDE disclaimer.

Page 1 of 25



Scientific Working Group on Digital Evidence

Individuals may not misstate or over represent duties and responsibilities of SWGDE work. This includes claiming oneself as a contributing member without actively participating in SWGDE meetings; claiming oneself as an officer of SWGDE without serving as such; claiming sole authorship of a document; use the SWGDE logo on any material or curriculum vitae.

Any mention of specific products within SWGDE documents is for informational purposes only; it does not imply a recommendation or endorsement by SWGDE.



Scientific Working Group on Digital Evidence

Table of Contents

1	Purpose	5
2	Scope	5
3	Limitations	5
4	Extraction and Analysis	5
4.1	Arduino.....	6
4.2	UART.....	9
4.3	SPI.....	13
4.4	Chip-off Analysis	23
5	Automation.....	24



Scientific Working Group on Digital Evidence

SWGDE Test Method for Bluetooth® Module Extraction and Analysis

Table of Figures

Figure 1. Example of a Host Controller (HC) 05 Bluetooth® module pinout.....	6
Figure 2. Bluetooth® HC-05 module with soldered leads.....	7
Figure 3: Bluetooth® module connected to development board and breadboard	7
Figure 4. AT commands and responses	8
Figure 5. USB to UART Bridge	9
Figure 6. Connections to RN	9
Figure 7. Connection between module and computer via COM8	10
Figure 8. Terminal Setup configuration.....	11
Figure 9. Log Settings.....	11
Figure 10. Adjusting baud rate.....	12
Figure 11. Placing the module in Command Mode	12
Figure 12. Example of the “D” Get command.....	13
Figure 13. Module’s response from the “Y” command.....	14
Figure 14. Module response showing firmware version.....	14
Figure 15. USB driver installation.....	15
Figure 16. RN42 module pinout	16
Figure 17. FT232 with soldered wires	17
Figure 18. Module connected to FT232 through breadboard	17
Figure 19. Selecting FT232 in BlueFlash	18
Figure 20. Successful connection to BlueFlash	18
Figure 21. Saving extraction	19
Figure 22. Pairing name.....	20
Figure 23. Mac address via SPI	21
Figure 24. MAC address of connected computer	22
Figure 25. MAC address via chip-off	23
Figure 26. Carving Example	24



Scientific Working Group on Digital Evidence

SWGDE Test Method for Bluetooth® Module Extraction and Analysis

1 Purpose

From IOT devices to credit card skimmers, Bluetooth® modules are used as a way to wirelessly connect devices. The modules contain identifiers of devices to which they were paired / connected. The purpose of this test method is to provide specific procedures required to recover those identifiers.

2 Scope

This document is intended for computer forensic practitioners conducting embedded device forensic examinations. Reference data is available, at a minimum, in the following publications:

- *SWGDE Best Practices for Examining Magnet Card Readers;*
- *ASTM E3017-19 Standard Practice for Examining Magnetic Card Readers;*
- *SWGDE Test Method for Skimmer Analysis – Analog Devices;*
- *SWGDE Test Method for Skimmer Analysis – Digital Devices;*
- *SWGDE Best Practices for Chip Off;*
- *SWGDE Best Practices for Computer Forensics Acquisitions; and*
- *SWGDE Embedded Device Core Competencies.*

3 Limitations

Bluetooth® Modules present unique examination challenges due to:

- Rapid changes in technology;
- Multiple data formats; and
- Lack of commercially available software to analyze data extracted from skimmers.

4 Extraction and Analysis

Bluetooth® module data may be extracted and analyzed in a few different ways. The process selected by the examiner will be module dependent. However, the first step to a successful examination is to locate information for the particular module that includes a pinout of external connections (see **Figure 1**).



Scientific Working Group on Digital Evidence

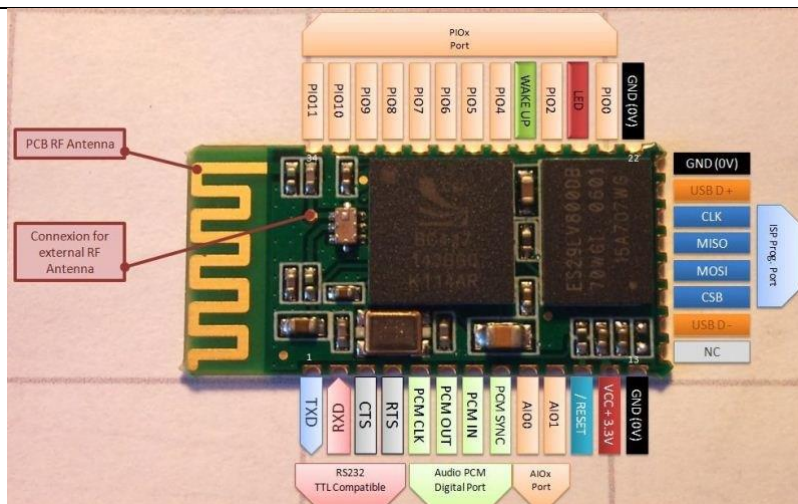


Figure 1. Example of a Host Controller (HC) 05 Bluetooth® module pinout

Once the pinout data is known, a development board is chosen. The development board will act as an interface between the module and the examination computer facilitating the transfer of data.

4.1 Arduino

Arduino is an open source development board that allows for small amounts of software, termed a “sketch” to be uploaded to the Arduino in order to accomplish desired tasks, in this case gather information from a Bluetooth® module. The sketch also contains the mechanical connection instructions and terminal settings instructions such as baud rate in addition to the programming code; a new or edited sketch can be created by the examiner if necessary, e.g. unsupported module. Once connected from the module to the Arduino and from the Arduino to a computer via USB, the examiner uses terminal software to transmit commands and receive data over the serial communications port established between the development board and the examination computer.

Arduino Integrated Development Environment (IDE) Version 1.8.2 is the terminal software used in this example, other serial terminals include PuTTY, RealTerm, and Tera Term. As there is no logging function available within Arduino IDE, the examiner should use screen capture software to record the session.

From within the Arduino software, the examiner:

- 4.1.1 Opens the sketch that matches the Bluetooth module;
- 4.1.2 Matches the com setting as reported by the operating system device manager;
- 4.1.3 Uploads the sketch from the examination computer to the Arduino.
- 4.1.4 The examiner then makes the physical connections to the transmission, reception, ground, and power (including power to the Key pin when present) pins on the module. If power is not applied to a present Key pin, the device powers to broadcast mode, i.e. mode that enables pairing. Only when the examiner applies power to both the VCC (power supply pin) and Key pins, does the module enter “Command Mode”, the mode needed to later communicate with the device without overwriting former pairing data. Once in Command Mode, the green light on the module will

SWGDE Test Method for Bluetooth® Module Extraction and Analysis

Version: 1.1 (September 17, 2020)

This document includes a cover page with the SWGDE disclaimer.

Page 6 of 25



Scientific Working Group on Digital Evidence

blink approximately every two seconds (versus a rapid flashing of the light which signifies broadcast mode).

In the following example, the examiner is instructed via a sketch to connect the module to a development board (Arduino® Uno R3) by connecting leads as follows:

- 4.1.4.1 Module transmit data (TXD) to Arduino® pin 2;
- 4.1.4.2 Module receive data (RXD) to Arduino® pin 3;
- 4.1.4.3 Module Key (PI011) to Arduino® pin 4;
- 4.1.4.4 Module VCC to Arduino pins 5 and 6 (multiple connections to a single pin can be facilitated with the use of a breadboard) to Arduino® pins 5 & 6;
- 4.1.4.5 Module ground to Arduino® ground.

See **Figure 2** and **Figure 3** for example connections.

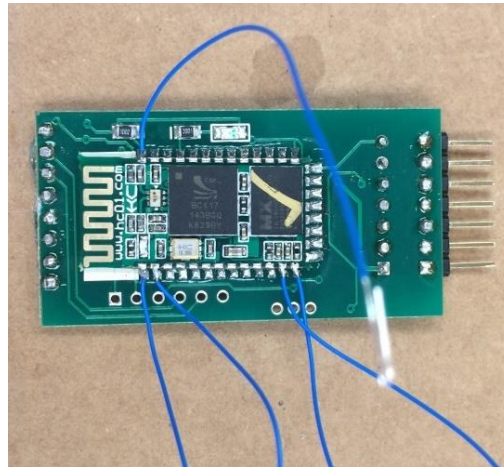


Figure 2. Bluetooth® HC-05 module with soldered leads

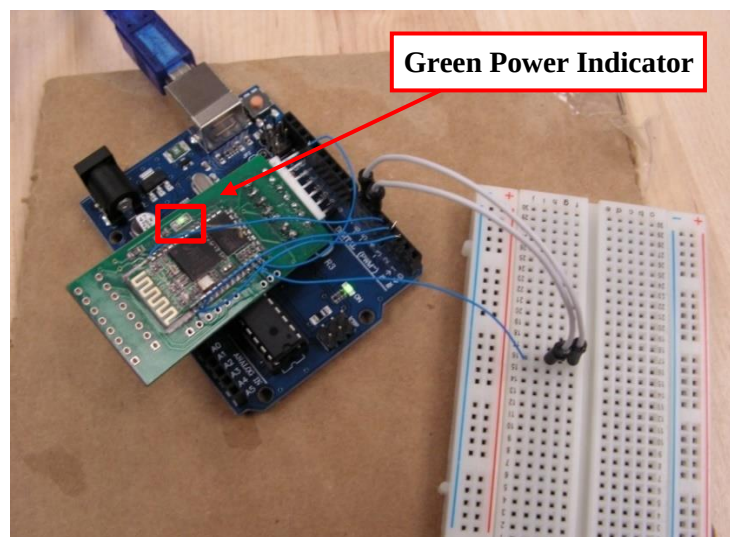


Figure 3: Bluetooth® module connected to development board and breadboard

SWGDE Test Method for Bluetooth® Module Extraction and Analysis

Version: 1.1 (September 17, 2020)

This document includes a cover page with the SWGDE disclaimer.

Page 7 of 25



Scientific Working Group on Digital Evidence

- 4.1.5 Within the software, the examiner opens the terminal and confirms a valid Command Mode serial connection by typing “AT” in the serial command box. If the device replies “OK” then the command mode serial connection is valid.
- 4.1.6 Once connected, the examiner can use the following commands to collect module artifacts. See **Figure 4** for examples of AT commands and responses.
- 4.1.6.1 “AT+VERSION?” – Responds with software version
 - 4.1.6.2 “AT+ADDR?” – Responds with the Bluetooth® module address
 - 4.1.6.3 “AT+NAME?” – Responds with the Bluetooth® name (default is HC-05)
 - 4.1.6.4 “AT+ROLE?” – Responds with device role (“0” is slave, “1” is master)
 - 4.1.6.5 “AT+PSWD?” – Responds with pairing code (default is 1234)
 - 4.1.6.6 “AT+ADCN?” – Responds with the authenticated device count (i.e. 6 or 8)
 - 4.1.6.7 “AT+MRAD?” – Responds with the Bluetooth® address of the most recently authenticated device
 - 4.1.6.8 “AT+FSAD=” – One can search for an authenticated device in the paired list by entering a known Bluetooth® address (“OK” = success or “FAIL” = failure)

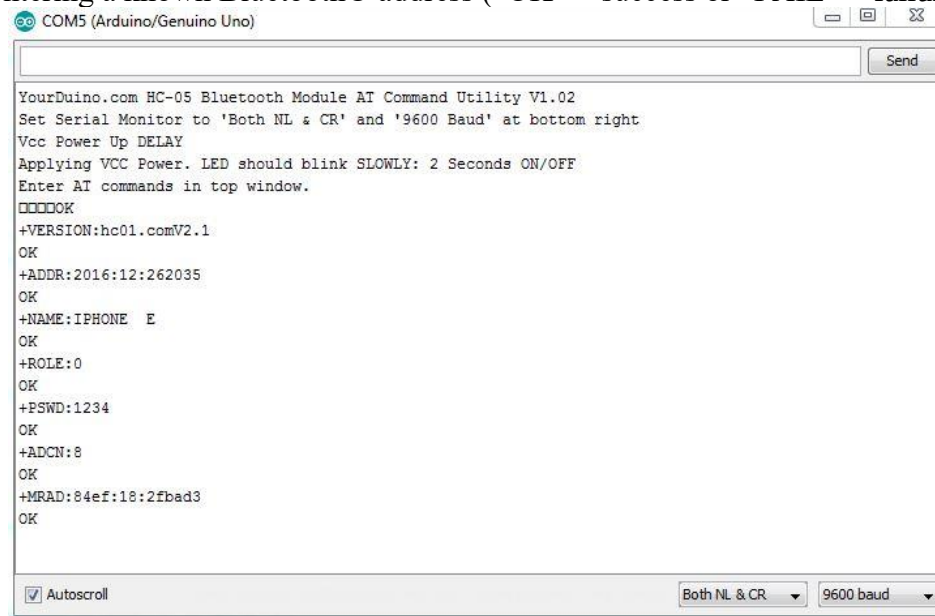


Figure 4. AT commands and responses

- 4.1.7 The queries and responses are captured and entered into the examiner’s report.



Scientific Working Group on Digital Evidence

4.2 UART

Different Bluetooth® modules may require different command sets. Using the example of a Roving Networks™, RN-41/42, a different interface to the examination computer and a different set of commands can be used. Instead of the Arduino used in the above example, the examiner may use an interface such as the USB to UART bridge pictured below.



Figure 5. USB to UART Bridge

- 4.2.1 To begin, the examiner sources a pin out diagram of the module, identifies and connects leads to the power, ground, TX (transmission) and RX (receive) pads. In order to make the soldered connections stronger, one may add hot glue.

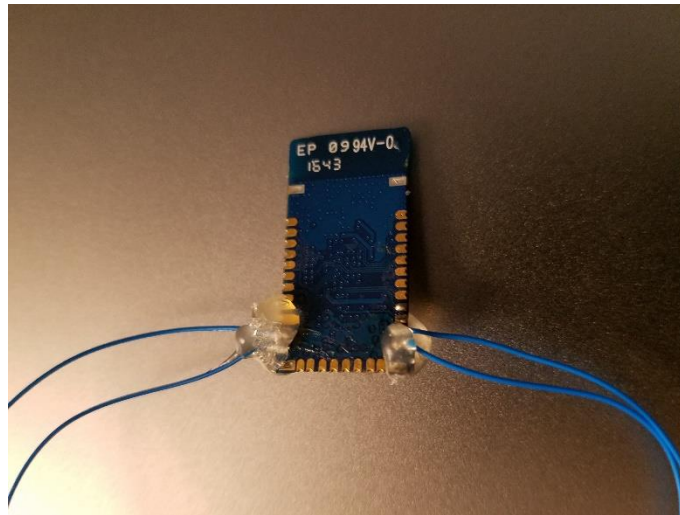


Figure 6. Connections to RN



Scientific Working Group on Digital Evidence

- 4.2.2 The examiner then connects the power and ground leads from the module to the interface board; however, TX from the module is connected to RX on the board, RX from the module is connected to TX on the board.
- 4.2.3 The examiner connects the interface board to his/her computer.
- 4.2.4 The examiner notes within the operating system's device settings, e.g. Device Manager within a Windows® operating system, what serial communication port is assigned to the interface bridge.
- 4.2.5 The examiner starts the development software, in this example Tera Term™, initiates a new connection, and selects the matching communication port (see **Figure 7**).

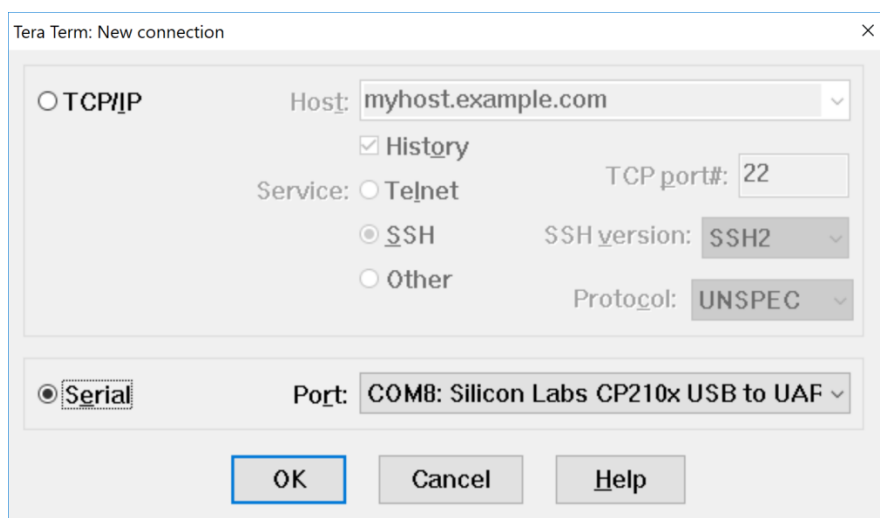


Figure 7. Connection between module and computer via COM8

- 4.2.6 Within “Terminal Setup” (see **Figure 8**) the examiner chooses CR (carriage return) and LF (line feed) for receiving and transmitting. This is done so the terminal will move the cursor down and to the beginning of the next line every time the ‘enter’ key is pressed. This is needed in order to send a command to the device. Additionally, the examiner chooses “Local echo” in order for the terminal to print to the screen what the examiner types. This is needed in order to verify commands are being typed correctly.



Scientific Working Group on Digital Evidence

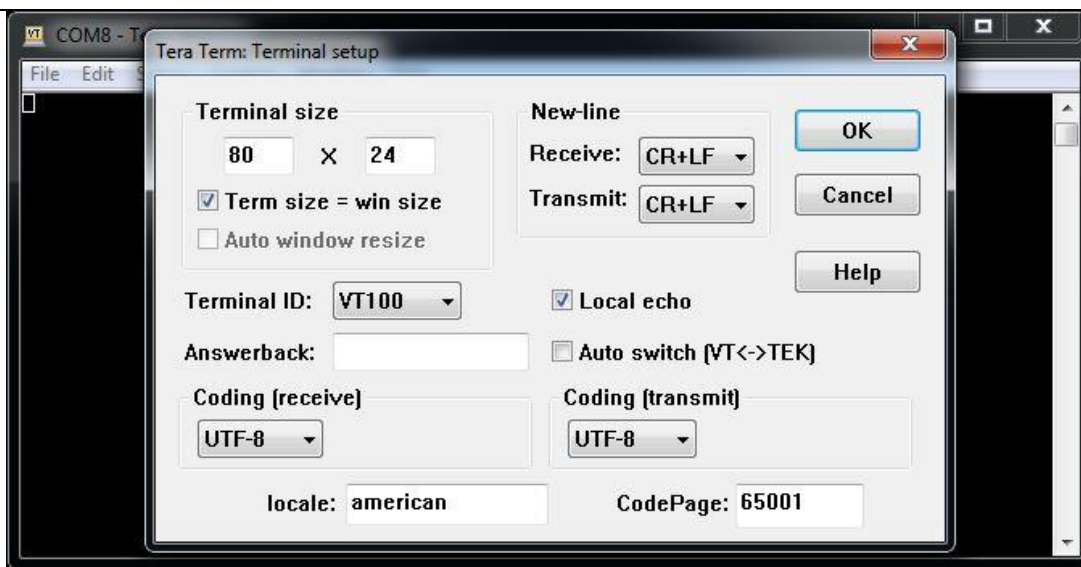


Figure 8. Terminal Setup configuration

- 4.2.7 The examiner then ensures logging is turned on by configuring the settings and providing a path to an output file.

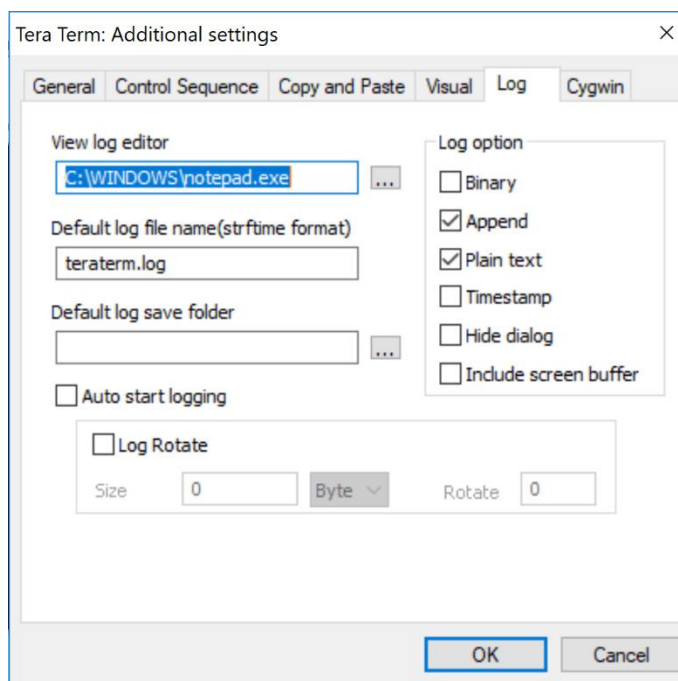


Figure 9. Log Settings



Scientific Working Group on Digital Evidence

- 4.2.8 The examiner then confirms the correct serial port communication speed is set by adjusting the baud rate if necessary. In this example, the correct rate is 9600.

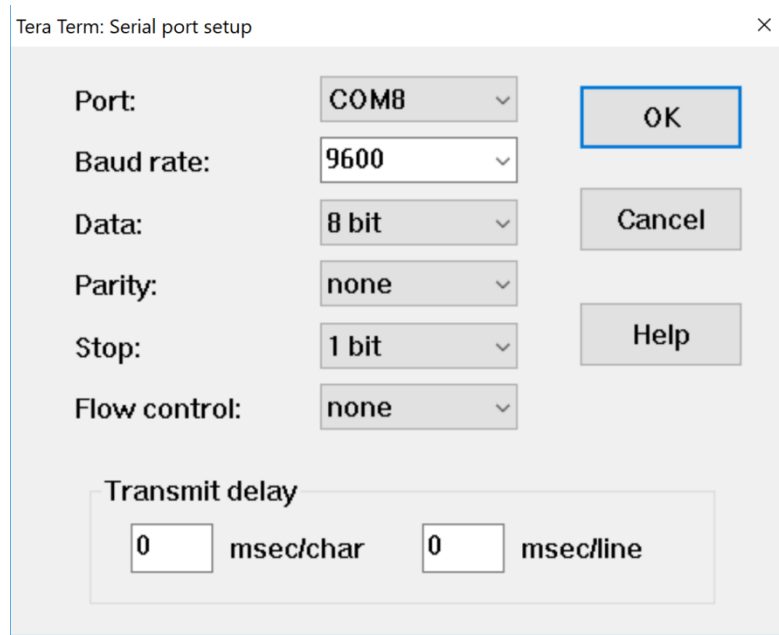


Figure 10. Adjusting baud rate

- 4.2.9 The examiner then places the module into command mode by entering “\$\$\$” into the terminal window. If successful, the terminal will respond, “CMD” (see **Figure 11**).

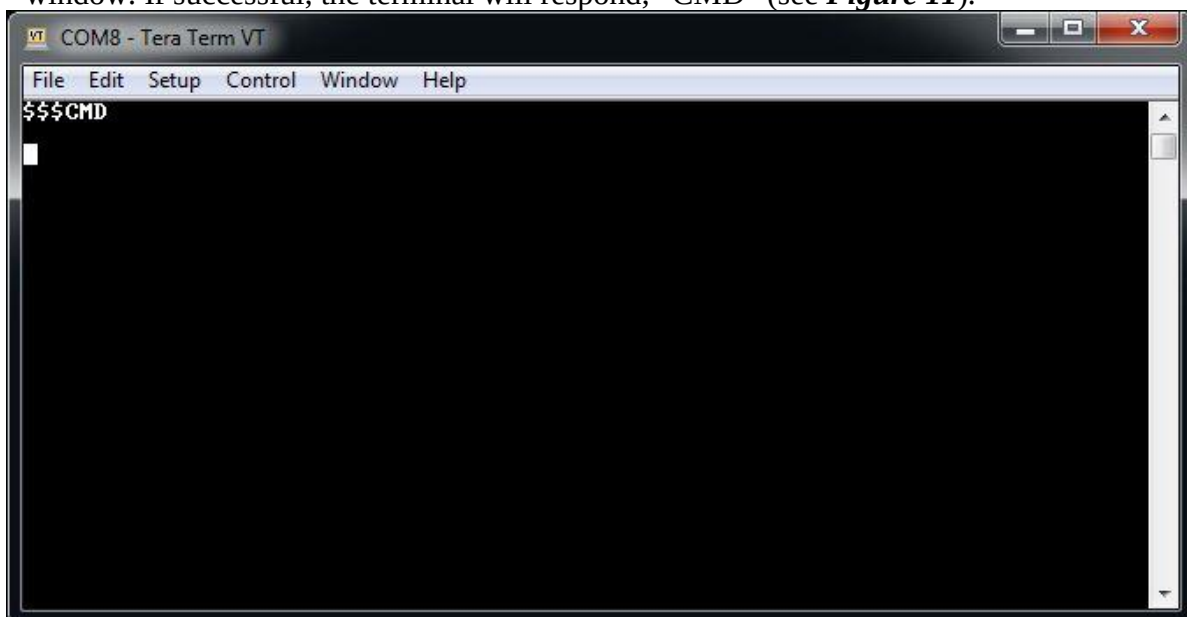


Figure 11. Placing the module in Command Mode



Scientific Working Group on Digital Evidence

4.2.10 The examiner continues by entering “Get” commands. A sample of commands and what they retrieve is as follows:

- 4.2.10.1 “D” – Displays basic settings such as the device Bluetooth® address, name, universal asynchronous receiver/transmitter (UART) settings, security, pin code, bonding, and remote address (Default settings are slave mode and pin code 1234, see Figure 12.)
- 4.2.10.2 “E” – Displays the extended settings such as the service name, service class, device class, and configuration timer
- 4.2.10.3 “GB” – Returns the device’s Bluetooth® address
- 4.2.10.4 “GF” – Returns the Bluetooth® address of the currently connected device
- 4.2.10.5 “GR” – Returns the stored remote Bluetooth address for re-connecting (if set)
- 4.2.10.6 “GK” – Returns the device’s current connection status (“1,0,0” indicates the device is connected; “0,0,0” indicates the device is not connected.)
- 4.2.10.7 “G<char>” – Displays the stored settings for a set command, where <char> is a set command name
- 4.2.10.8 “\$\$\$” – Causes the device to enter command mode, displaying “CMD”
- 4.2.10.9 “---” – Causes the device to exit command mode, displaying “END”
- 4.2.10.10 “H” – Displays a list of commands and their basic syntax
- 4.2.10.11 “V” – Displays the firmware version

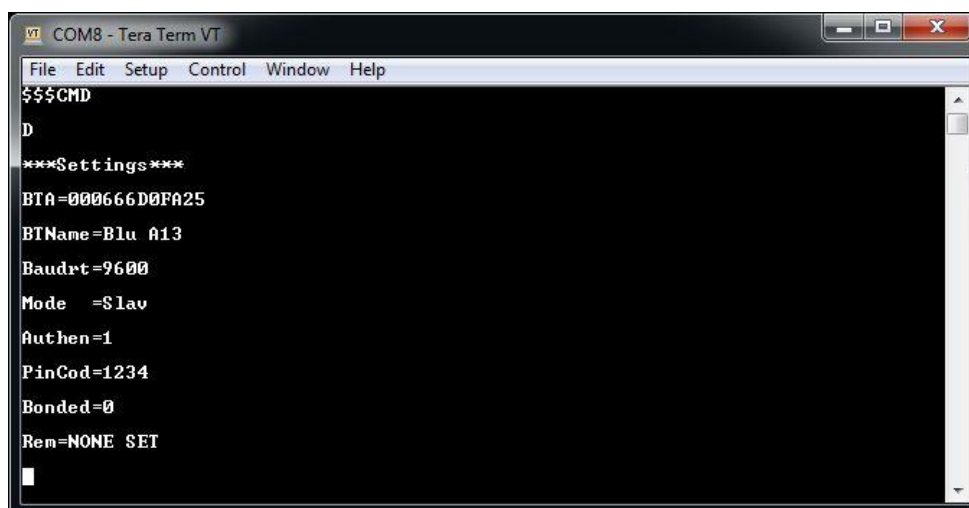


Figure 12. Example of the “D” Get command

4.2.11 The examiner then inserts the log of the queries and module responses within his/her report.

4.3 SPI

In the example above, one may notice that a possibly important artifact is not available. The last connected device (MAC address) is not listed. This is because for this particular module, the command that would have pulled that artifact (the “Y” command) is not available via UART until firmware version 6.3. Unfortunately, RN modules with firmware version 6.15 are commonly used. The “GF”



Scientific Working Group on Digital Evidence

command will only provide the last connected MAC until a module power cycle. See Figure 37 and Figure 38 for responses to the “Y” and “V” commands.

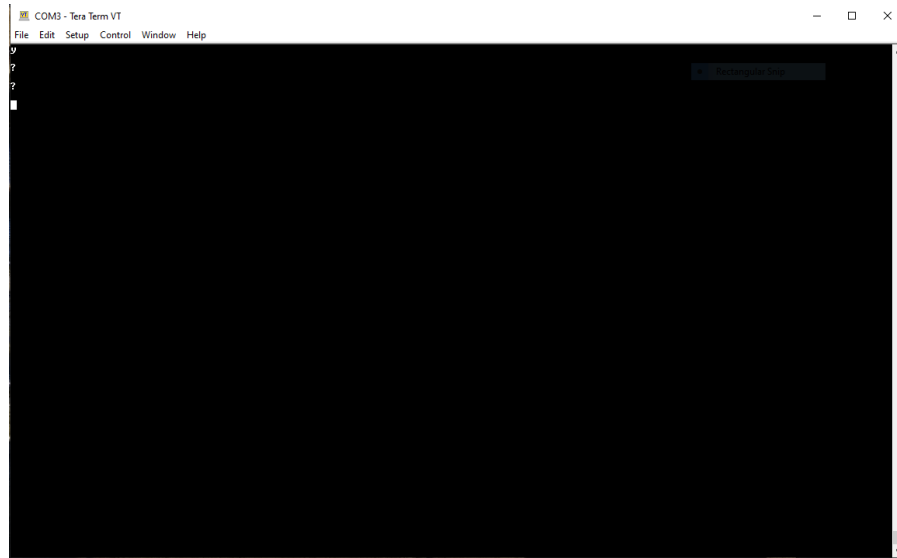


Figure 13. Module’s response from the “Y” command

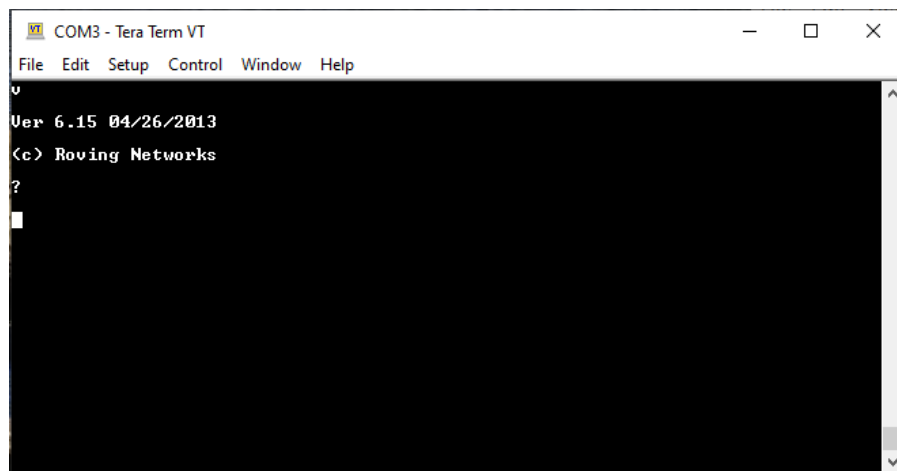


Figure 14. Module response showing firmware version



Scientific Working Group on Digital Evidence

Another option for the examiner is to extract information from the module using the SPI protocol. Data extracted via SPI, or through chip-off (covered later in this document), allows for the analysis of a write protected binary image.

4.3.1 SPI Extraction

In order to complete an extraction via SPI, the examiner must first source and load SPI libraries onto a FT232 serial adapter board. This can be accomplished through the use of a USB driver installation tool, such as Zadig, pictured in **Figure 15**.

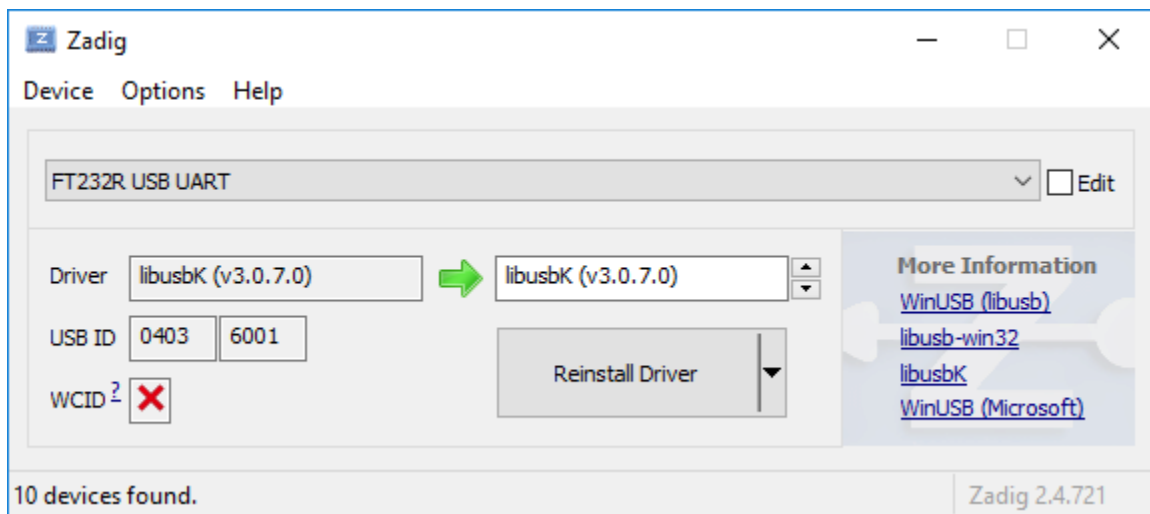


Figure 15. USB driver installation

4.3.1.1 The examiner connects the FT232 board to an examination computer, then within the Zadig software, chooses the FT232 board in the top drop down menu, uses the up/down arrows to choose the libusbK (v3.0.7.0) driver in the right side of the window, and presses “Reinstall Driver”.

4.3.2 With the FT232 board prepared, the examiner’s next step is to source the pinout for the Bluetooth® module to be examined; in this example, the Roving Networks RN42. The pinout for the RN42 with highlighted pins that correspond with the SPI protocol for data extraction is



Scientific Working Group on Digital Evidence

pictured in **Figure 16**. Note: the pin labelling is shown from the top while the mechanical connections are made to the pads underneath the module.

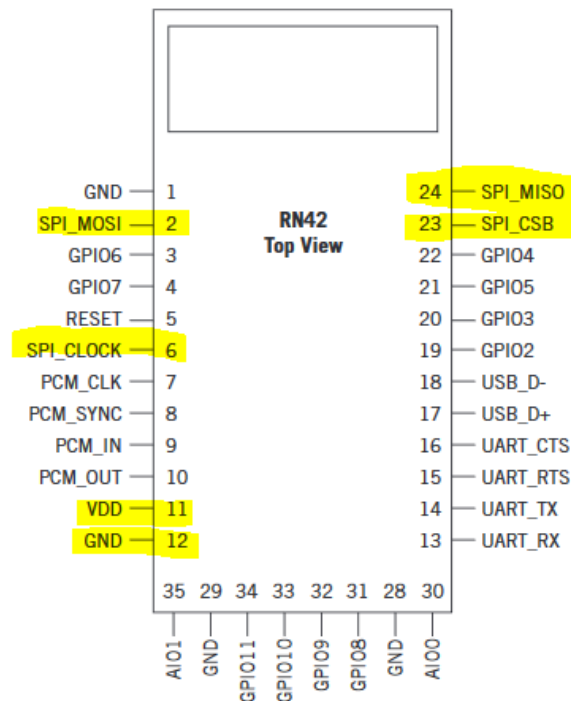


Figure 16. RN42 module pinout

- 4.3.3 Continuing, the examiner must match the SPI pins (plus power and ground) of the module to the correct pins on the FT232. The connections are as follows:

DSR/RSD = MISO
RI = MOSI
RTS = CLK
DTR = CSB
3.3V = VDD
GND = GND

Figure 17 shows wires soldered to the appropriate FT232 board pads. Note: The jumper should be set to 3.3. volts. The alternative, 5 volts, could damage the Bluetooth® module.



Scientific Working Group on Digital Evidence

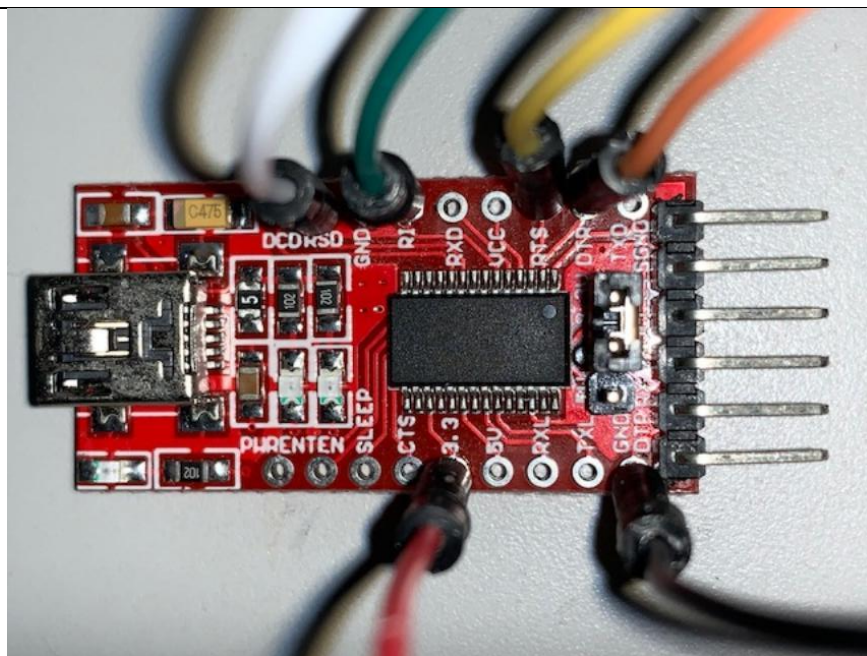


Figure 17. FT232 with soldered wires

A bread board may be used to facilitate the connections between the two wire sets. Doing so relieves some of the torque that is placed on the pads that occurs if the wires are connected directly from module to FT232 board.

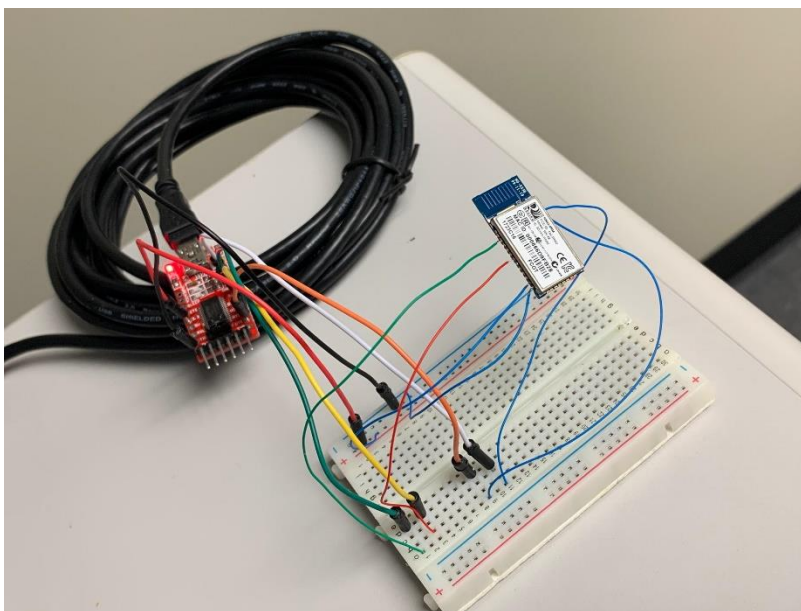


Figure 18. Module connected to FT232 through breadboard



Scientific Working Group on Digital Evidence

- 4.3.4 With the module connected to the FT232 board through a breadboard and the FT232 board connected via USB to an examination computer, the examiner then uses a software application to extract the module's contents. In this example, a program called BlueFlash is used.
- 4.3.4.1 Using BlueFlash, the examiner chooses the FT232 board using the drop down arrow. This is shown in **Figure 19**.

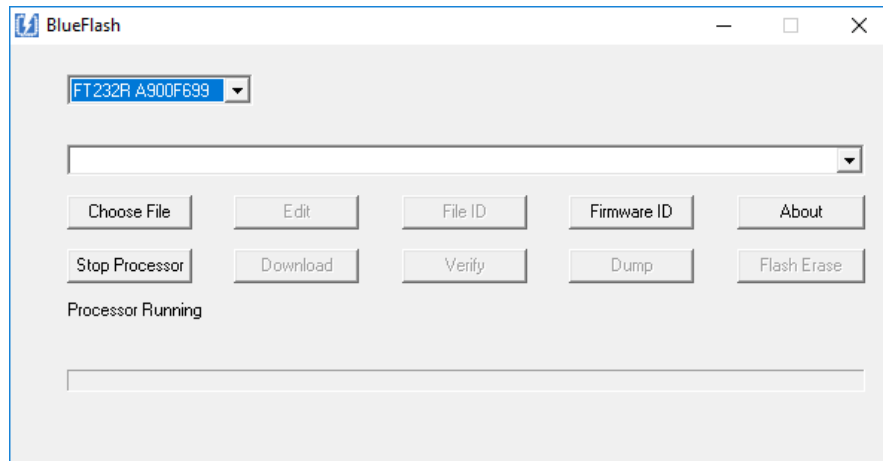


Figure 19. Selecting FT232 in BlueFlash

- 4.3.4.2 Once selected, BlueFlash will attempt to recognize the Bluetooth® module's processor. If successful, BlueFlash will report that the module is recognized and will provide size data of the module's flash storage. The examiner then selects "Stop Processor". This step is required for the software to complete a read of the module's flash chip. With the processor stopped, the examiner hits the "Dump" button. A screenshot of the BlueFlash software showing these available selections is shown in **Figure 20**.

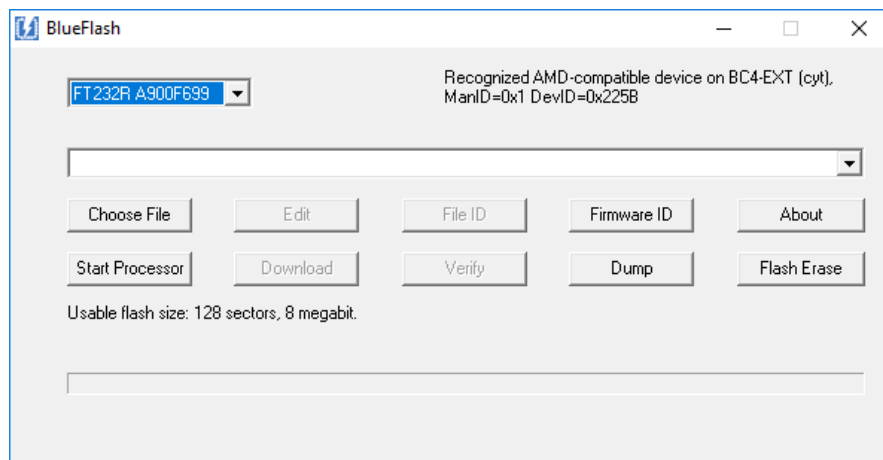


Figure 20. Successful connection to BlueFlash



Scientific Working Group on Digital Evidence

- 4.3.4.3 The examiner is given the option of saving the extraction in a couple of formats. To facilitate efficient carving, the examiner should use the “Raw files” option in order to save the extraction as a binary file. The process for saving the extraction as a binary file is shown below in **Figure 21**.

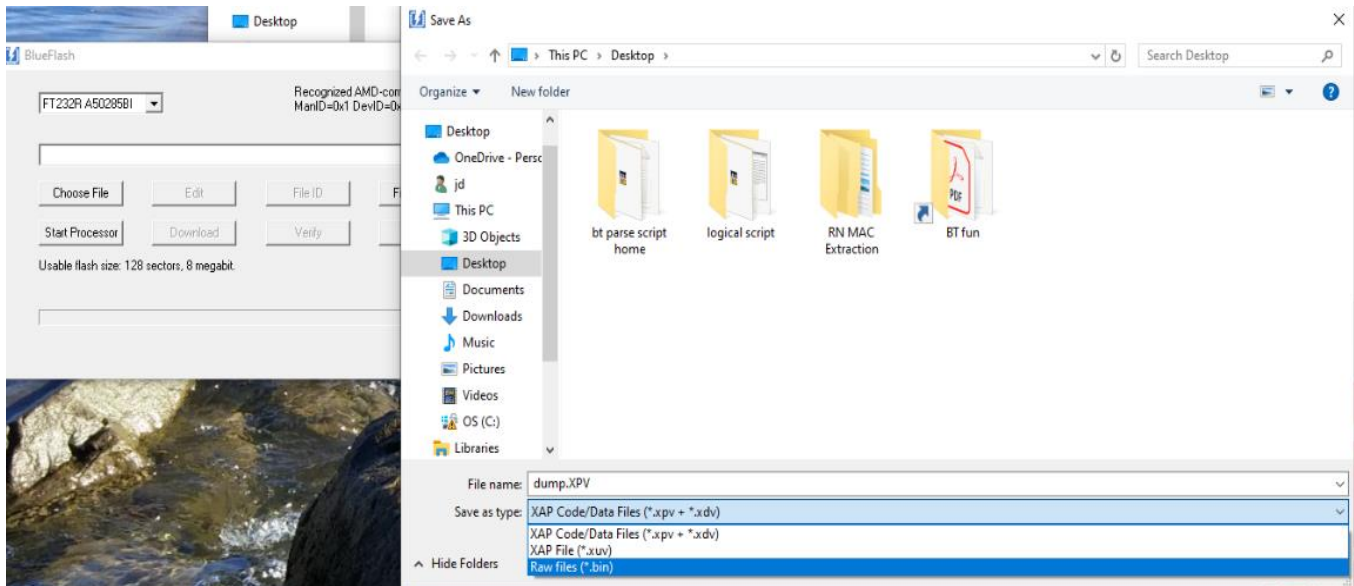


Figure 21. Saving extraction

4.3.5 SPI Extraction Analysis

Regarding analysis of the extraction, artifacts such as the Bluetooth® pairing name is observable when opening the extraction in a hex editor. A pairing name, RNBT – FB28, in 8bit ASCII/Unicode is shown in **Figure 22**.



Scientific Working Group on Digital Evidence

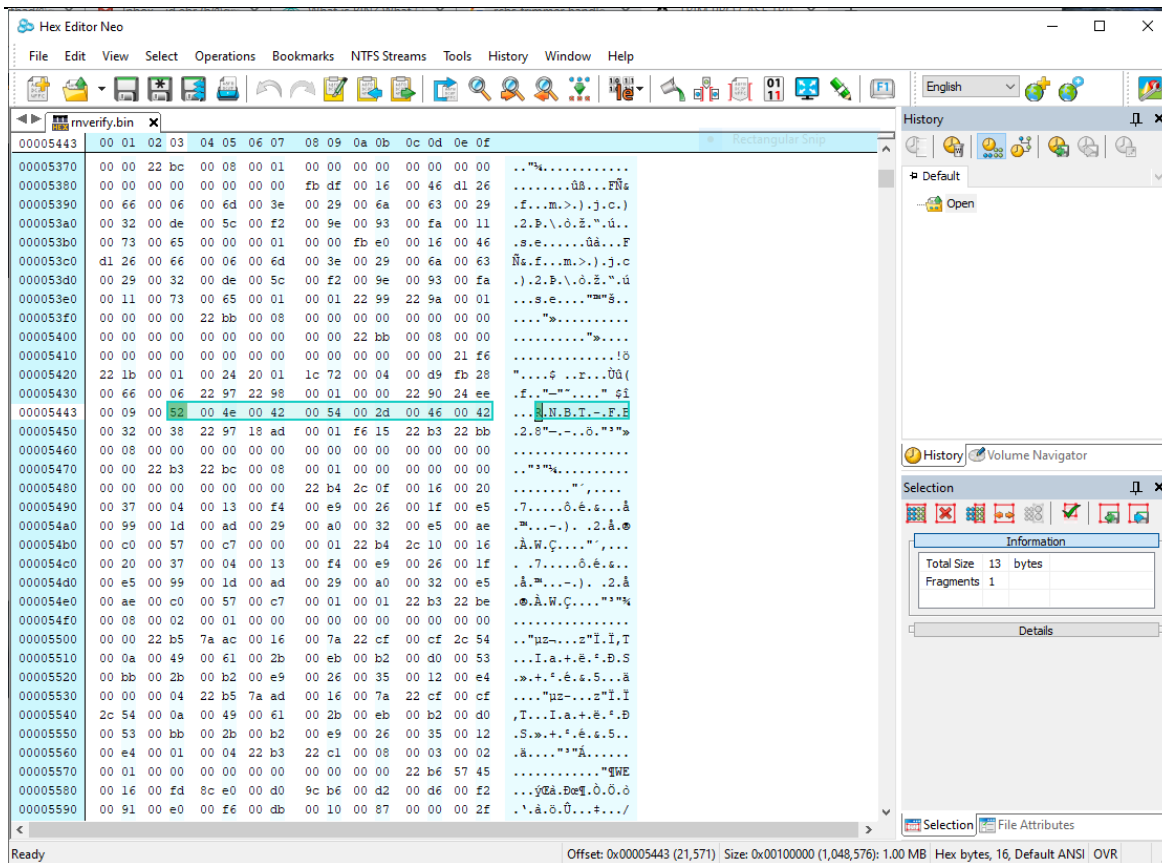


Figure 22. Pairing name



Scientific Working Group on Digital Evidence

4.3.6 The recovery of connected Bluetooth® addresses is more complicated. **Figure 23** shows a MAC address, but it is encoded in a way that is not immediately apparent.

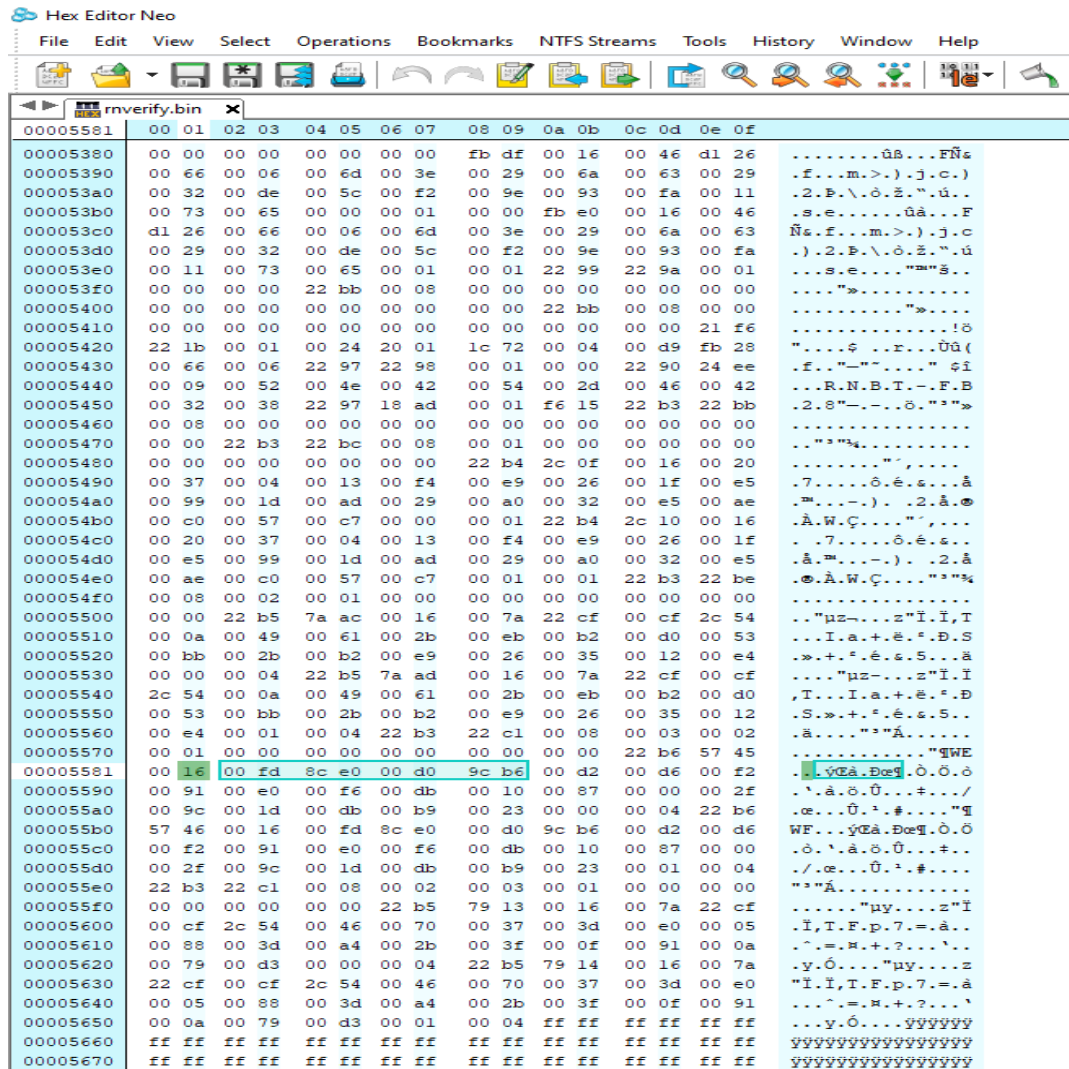


Figure 23. Mac address via SPI

4.3.7 The hexadecimal values highlighted above: 16 00 fd 8c e0 00 d0 9c b6, represent the MAC address: 9c b6 d0 fd 8c e0, as confirmed in the test computer that paired with the module. The test computer's network properties are displayed in **Figure 24**.



Scientific Working Group on Digital Evidence

← Settings

View your network properties

Network category:	Home
Connectivity (IPv4/IPv6):	Connected to Internet / Connected to unknown network
Name:	Bluetooth Network Connection 2
Description:	Bluetooth Device (Personal Area Network) #2
Physical address (MAC):	9c:b6:d0:fd:8c:e0
Status:	Not operational
Maximum transmission unit:	1500
IPv4 address:	169.254.243.79/16
IPv6 address:	fe80::5c69:c7f2:a95b:f34f%6/64
DNS servers:	fec0:0:0:ffff::1%1, fec0:0:0:ffff::2%1, fec0:0:0:ffff::3%1
Connectivity (IPv4/IPv6):	Disconnected

Figure 24. MAC address of connected computer

- 4.3.8 The format for the extracted address, 16 00 fd 8c e0 00 d0 9c b6, is as follows:
- 1st byte, 0x16 = serves as a header
 - 2nd byte, 0x00 = padding
 - 3rd byte, 0xfd = the fourth byte of the address
 - 4th byte, 0x8c = the fifth byte of the address
 - 5th byte, 0xe0 = the sixth byte of the address
 - 6th byte, 0x00 = padding
 - 7th byte, 0xd0 = the third byte of the address
 - 8th byte, 0x9c = the first byte of the address
 - 9th byte, 0xb6 = the second byte of the address
- Resolving to the MAC address of 9c:b6:d0:fd:8c:e0.



Scientific Working Group on Digital Evidence

The encoding uses the Bluetooth® address structure:

2 bytes of Non-significant Address Part (NAP);

1 byte of Upper Address Part (UAP); and

3 bytes of Lower Address Part (LAP).

However, organized as LAP, UAP, and NAP and in Big Endian.

4.4 Chip-off Analysis

While the pairing name is still viewed in close proximity to paired device MAC address, the formatting of the Bluetooth® address is a little different. **Figure 25** shows data from the same module but from a chip-off extraction.

Address	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00005580	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00005360	25	00	00	00	8d	f4	04	00	46	00	26	d1	66	00	06	00
00005370	00	00	bc	22	08	00	01	00	00	00	00	00	00	00	00	00
00005380	00	00	00	00	00	00	00	00	df	fb	16	00	46	00	26	d1
00005390	66	00	06	00	6d	00	3e	00	29	00	6a	00	63	00	29	00
000053a0	32	00	de	00	5c	00	f2	00	9e	00	93	00	fa	00	11	00
000053b0	73	00	65	00	00	00	01	00	00	00	e0	fb	16	00	46	00
000053c0	26	d1	66	00	06	00	6d	00	3e	00	29	00	6a	00	63	00
000053d0	29	00	32	00	de	00	5c	00	f2	00	9e	00	93	00	fa	00
000053e0	11	00	73	00	65	00	01	00	01	00	99	22	9a	22	01	00
000053f0	00	00	00	00	bb	22	08	00	00	00	00	00	00	00	00	00
00005400	00	00	00	00	00	00	00	00	00	00	bb	22	08	00	00	00
00005410	00	00	00	00	00	00	00	00	00	00	00	00	00	00	f6	21
00005420	1b	22	01	00	24	00	01	20	72	1c	04	00	d9	00	28	fb
00005430	66	00	06	00	97	22	98	22	01	00	00	00	90	22	ee	24
00005440	09	00	52	00	4e	00	42	00	54	00	2d	00	46	00	42	00
00005450	32	00	38	00	97	22	ad	18	01	00	15	f6	b3	22	bb	22
00005460	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00005470	00	00	b3	22	bc	22	08	00	01	00	00	00	00	00	00	00
00005480	00	00	00	00	00	00	00	00	b4	22	0f	2c	16	00	20	00
00005490	37	00	04	00	13	00	f4	00	e9	00	26	00	1f	00	e5	00
000054a0	99	00	1d	00	ad	00	29	00	a0	00	32	00	e5	00	ae	00
000054b0	c0	00	57	00	c7	00	00	00	01	00	b4	22	10	2c	16	00
000054c0	20	00	37	00	04	00	13	00	f4	00	e9	00	26	00	1f	00
000054d0	e5	00	99	00	1d	00	ad	00	29	00	a0	00	32	00	e5	00
000054e0	ae	00	c0	00	57	00	c7	00	01	00	01	00	b3	22	be	22
000054f0	08	00	02	00	01	00	00	00	00	00	00	00	00	00	00	00
00005500	00	00	b5	22	ac	7a	16	00	7a	00	cf	22	cf	00	54	2c
00005510	0a	00	49	00	61	00	2b	00	eb	00	b2	00	d0	00	53	00
00005520	bb	00	2b	00	b2	00	e9	00	26	00	35	00	12	00	e4	00
00005530	00	00	04	00	b5	22	ad	7a	16	00	7a	00	cf	22	cf	00
00005540	54	2c	0a	00	49	00	61	00	2b	00	eb	00	b2	00	d0	00
00005550	53	00	bb	00	2b	00	b2	00	e9	00	26	00	35	00	12	00
00005560	e4	00	01	00	04	00	b3	22	c1	22	08	00	03	00	02	00
00005570	01	00	00	00	00	00	00	00	00	00	00	00	b6	22	45	57
00005580	16	00	fd	00	e0	8c	d0	00	b6	9c	d2	00	d6	00	f2	00
00005590	91	00	e0	00	f6	00	db	00	10	00	87	00	00	00	2f	00
000055a0	9c	00	1d	00	db	00	b9	00	23	00	00	00	04	00	b6	22
000055b0	46	57	16	00	fd	00	e0	8c	d0	00	b6	9c	d2	00	d6	00
000055c0	f2	00	91	00	e0	00	f6	00	db	00	10	00	87	00	00	00
000055d0	2f	00	9c	00	1d	00	db	00	b9	00	23	00	01	00	04	00
000055e0	b3	22	c1	22	08	00	02	00	03	00	01	00	00	00	00	00
000055f0	00	00	00	00	00	00	b5	22	13	79	16	00	7a	00	cf	22
00005600	cf	00	54	2c	46	00	70	00	37	00	3d	00	e0	00	05	00
00005610	88	00	3d	00	a4	00	2b	00	3f	00	0f	00	91	00	0a	00

Figure 25. MAC address via chip-off



Scientific Working Group on Digital Evidence

4.4.1 The format for the address, 16 00 fd 00 e0 8c d0 00 b6 9c, is as follows:

- 1st byte, 0x16 = serves as a header
 - 2nd byte, 0x00 = padding
 - 3rd byte, 0xfd = the fourth byte of the address
 - 4th byte, 0x00 = padding
 - 5th byte, 0xe0 = the sixth byte of the address
 - 6th byte, 0x8c = the fifth byte of the address
 - 7th byte, 0xd0 = the third byte of the address
 - 8th byte, 0x00 = padding
 - 9th byte, 0xb6 = the second byte of the address
 - 10th byte, 0x9c = the first byte of the address
- Resolving to the MAC address of 9c:b6:d0:fd:8c:e0.

Once again, the extraction shows the module in the LAP UAP NAP; however, it is presented as Little Endian.

5 Automation

As data resident on the tested modules included addresses, pairing codes, and broadcast names not entirely from seeding events, the Endian-ness, record identifiers, and device specific headers, e.g. 0x16 for an RN module, must be factored into any artifact carving solution. Similar to other devices, e.g. drones, it is presumed that manufacturers seed data to test function before shipping to distributors.

Figure 26 shows an example of a scripted carving solution.

```
Windows PowerShell

BlueTooth Module Carver v3

Possible module MAC address:
0x542a 00:06:66:D9:FB:28 (Roving Networks)

Possible Bluetooth pairing pin(s):
0x4092: 1234

Possible broadcast name(s):
0x543c: R N B T - F B 2 8

Possible paired MAC address(es):
0x5530: 2C:54:CF:7A:22:CF (LG Electronics (Mobile Communications))
0x55aa: 9C:B6:D0:FD:8C:E0 (Rivet Networks)
0x5624: 2C:54:CF:7A:22:CF (LG Electronics (Mobile Communications))
```

Figure 26. Carving Example



Scientific Working Group on Digital Evidence

History

Revision	Issue Date	Section	History
Draft	01/22/2020	All	Initial draft for public comment.
DRAFT 1.0	01/22/2020	All	Formatted and technical edit performed for release as a Draft for Public Comment.
Draft 1.1	8/27/2020	All	Technical edits.
1.1	9/17/2020	All	Voted for release as final publication