

---

# sep100 S1 module usage

Graz, 01.12.2021

This document describes in how to wire and communicate with the sep100 S1 IoT module via UART.

## Contents

<b>USAGE .....</b>	<b>2</b>
Flashing .....	2
After successfully flashing the module be sure to reboot it, i.e. reconnect power supply or trigger reset via the reset pin (pull LOW). ....	2
Operation .....	2
<b>UART COMMANDS .....</b>	<b>3</b>
<b>EXAMPLE USAGE CHARTS .....</b>	<b>7</b>
<b>PRODUCT LABELING .....</b>	<b>9</b>
<b>RF EXPOSURE .....</b>	<b>10</b>
<b>SPURIOUS EMISSION TESTS .....</b>	<b>11</b>
<b>FCC COMPLIANCE .....</b>	<b>12</b>

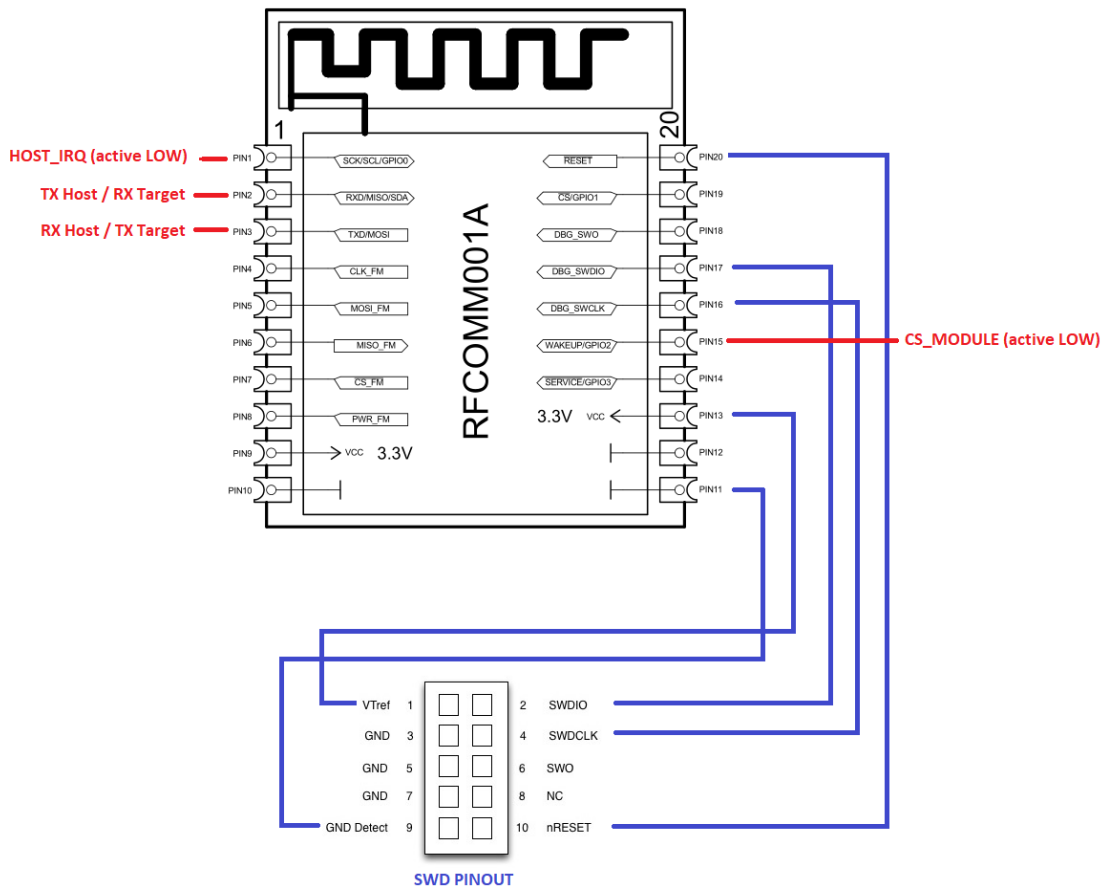


Figure 1RCOMM001A pinout

## Flashing

To flash the Module with the SEGGER Flasher (ARM) connect the module via the SWD pins shown in blue in Figure 1RCOMM001A pinout.

An external power supply (3.3V) is also needed as the SEGGER flasher does not provide power.

Start “ARMFirmwareFlasher.exe” and select a firmware hex file via the drop-down menu:

**“File – Open firmware file”**

Enter a Label ID (must match a Label ID in the pool-file.csv) and press **START**.

After a successful flash operation, the notification windows should turn green and **PASS** appears.

After successfully flashing the module be sure to reboot it, i.e. reconnect power supply or trigger reset via the reset pin (pull LOW).

## Operation

For communicating one must first pull PIN 15 (WAKEUP/GPIO2) low and wait for the “RDY\r” response from the module. After this the module is ready to receive commands for approximately two seconds. A timeout will occur if this time is exceeded.

All available commands a listed below including a short example and the expected result.

# UART COMMANDS

---

## **Command "RES\r":**

Function: resets the module

Returns: "ACK\r"

## **Command "SVN\r":**

Function: returns the SVN-revision of the firmware (like FW version)

Returns: "ACK 9403\r"

9403 => 0x0394 in decimal = 916 => SVN revision 916.

## **Command "FIB\r":**

Function: returns very detailed firmware information.

Returns: "ACK 0000000003200203100094038524A74A0020000000D001000000C002FFFF0BF1\r"

## **Command "RID\r":**

Function: returns the unique ID (=address) of the module.

Returns: "ACK 0012FFA5\r"

0012FFA5 => 0xA5FF1200 => the unique ID of the module is A5FF1200.

## **Command "SEV <1-byte event-type><4-byte data>\r":**

Function: sends a high-priority event to the access-point backend.

<1-byte event-type>:

specifies the "type" of the event (see PKT\_EVENT\_TYPES).

PKT\_EVENT\_TYPE\_NONE = 0x00, // no event

PKT\_EVENT\_TYPE\_BUTTON\_1\_PRESSED = 0x01, // event "button 1 was pressed"

PKT\_EVENT\_TYPE\_BUTTON\_2\_PRESSED = 0x02, // event "button 2 was pressed"

PKT\_EVENT\_TYPE\_BUTTON\_1\_AND\_2\_PRESSED = 0x03, // event "button 1+2 was pressed"

PKT\_EVENT\_TYPE\_SIMPLE\_NFC\_EVENT = 0x04

// ... add more event types here (up to 0xFF)

<4-byte data>: 4-byte payload.

Example: "SEV 04AABBCCDD\r"

Format: 04 => event type 0x04

AABBCCDD => 0xAA,0xBB,0xCC,0xDD payload that will be sent to the access-point.

Returns: "ACK\r"

Notes: Events are reserved for critical messages and should be used seldomly.

Sending too many events will cause congestion of the network,

therefore the use should be limited to high-priority or critical messages.

Only a single event can be active at a given time.

To monitor the status and successful/failed delivery, use the

"Get Event Status" command (EVS).

### **Command "EVS\r":**

Function: returns the status of the event-transmission.

Returns: "ACK 000A0104AABBCCDD\r"

Format:

offset 0: flag indicating if event is active

offset 1: number of retries left

offset 2: result of previous event (see SYNC\_EVENT\_RESULT)

SYNC\_EVENT\_RESULT\_NOT\_AVAILABLE = 0 // result is not available

SYNC\_EVENT\_RESULT\_EVENT\_CONFIRMED = 1 // event was confirmed

SYNC\_EVENT\_RESULT\_EVENT\_TIMEOUT = 2 // event has timed out

offset 3: event type (see PKT\_EVENT\_TYPES)

PKT\_EVENT\_TYPE\_NONE = 0x00 // no event

PKT\_EVENT\_TYPE\_BUTTON\_1\_PRESSED = 0x01 // event "button 1 was pressed"

PKT\_EVENT\_TYPE\_BUTTON\_2\_PRESSED = 0x02 // event "button 2 was pressed"

PKT\_EVENT\_TYPE\_BUTTON\_1\_AND\_2\_PRESSED = 0x03 // event "button 1+2 was pressed"

PKT\_EVENT\_TYPE\_SIMPLE\_NFC\_EVENT = 0x04

// ... add more event types here (up to 0xFF)

// offset 4-7: event data payload (4 bytes)

### **Command "RSC <1-byte user-SET-config ID>\r":**

Function: reads a user-SET-config register and returns the 12-bits data.

This config was SET by the access-point backend system.

<1-byte user-SET-config ID>: specifies the config-register to read.

Example: "RSC 00\r"

00 => 0x02 => use user-SET-config #0

Returns: "ACK BC0A\r"

BC0A => 0x0ABC => 12 bits of config data => 0xABC

Notes: Config-registers can be set/read the access-point backend system.

ATTENTION! Only 12 bits can be SET by the access-point!

However, 4 bytes can be READ by the access-point!

So, the RSC and WGC commands differ in their function!

### **Command "WGC <1-byte user-GET-config ID><4-byte data>\r":**

Function: writes a 4-byte value to a user-GET-config register.

<1-byte config-ID>: specifies the config-register to write.

<4-byte data>: 4 byte payload.

This config can be READ by the access-point backend system.

Note that the value will not be transferred to the access-point,

it's the access-point backend system's task to regularly read

all configs of interest. This method does not cause a congestion of the network and is

the preferred way of providing data to the access-point backend.

Example: "WGC 02AABBCCDD\r"

02 => 0x02 = use user-GET-config #1.

AABBCCDD => 0xAA,0xBB,0xCC,0xDD payload that will be written.

Returns: "ACK\r"

Notes: Config-registers can be set/read the the access-point backend system.

ATTENTION! 4 bytes can be READ by the access-point!

However only 12 bits can be SET by the access-point!

So, the RSC and WGC commands differ in their function!

### **Command "EHI <32bit host interrupt configs\r":**

Function: Enables interrupts to signal host about certain events.

Check "interrupt\_commands.py" for details about what interrupts to active and the according command string.

Example: "EHI 00010000\r"

"Host enables IRQ when [user-SET-config #0] will be written."

NOTE: this could to be configured after startup once.

Returns: "ACK\r"

### **Command "GHI\r":**

Function: Prints currently active IRQ flags.

Example: "GHI\r"

Returns: "ACK 00010000\r"

IRQ flag [user-SET-config #0 was written] is set.

### **Command "CHI\r":**

Function: Host wants to clear IRQ flag.

Example: "EHI 00010000\r"

host wants to clear IRQ flag, user-SET-config #0 was written."

Returns: "ACK\r"

Notes: After an interrupt was caught and handled, PIN 1 (GPIO0) is still low,

one must manually clear the active interrupt in order to reset PIN 1 to a HIGH state again.

To clear all active interrupts, use the "CHI FFFFFFFF\r" command which will clear everything.

### **Command "ERF\r":**

Function: Enables output of modulated or unmodulated carrier

Example: "ERF 0107\r"

Enabled output of unmodulated carrier on channel 7

Returns: "ACK\r"

Notes: The first bytes describe the mode :

(00 = disable radio; 01 = output unmodulated carrier; 02 = output modulated carrier)

The second bytes describes the channel to use (0x00 – 0x0B)

### **Command "RUD\r":**

Function: Reads USERDATA previously received via AUX page

Example: "RUD 00003100\r"

Reads 0x100 (256) bytes of USERDATA from address 0x00003

Returns: 2 bytes length + requested data + 2 bytes checksum

Notes: Every packet is transmitted with a two byte length information in the beginning and a checksum at the end for validating the requested data.

This command allows to read a maximum of 4095 bytes per request in address range of 0x00000 to 0xFFFFF (1048575), although an error will be thrown if the address is out of range.

### Error responses "ERR <2-byte error-code>\r":

Function: any function may respond with a 2-byte error code.

Example: "ERR 0200\r"

0200 => 0x0002 => ERR\_COMMAND\_TIMEOUT.

List of error-codes:

ERR_NONE	= 0 // no error - all OK!
ERR_UNSPECIFIC	= 1 // a generic error (not a specific one) occurred
ERR_COMMAND_TIMEOUT	= 2 // host-MCU didn't send a command on time (=timeout)
ERR_COMMAND_BUFFER_FULL	= 3 // buffer for receiving commands was full
ERR_COMMAND_RECEIVE_ABORTED	= 4 // host-MCU command reception was aborted
ERR_COMMAND_LENGTH_INVALID	= 5 // host-MCU command had invalid length
ERR_COMMAND_UNKNOWN	= 6 // command is unknown
ERR_PARAMETER_LENGTH_INVALID	= 7 // parameter had invalid length
ERR_PARAMETER_FORMAT_INVALID	= 8 // parameter had invalid format
ERR_COMMAND_NOT_IMPLEMENTED	= 9 // command is not implemented
ERR_EVENT_IS_ACTIVE	= 10 // an EVENT is still active
ERR_USER_CONFIG_ID_INVALID	= 11 // user-config ID is invalid
ERR_RFOUT	= 12, // rf output config is invalid
ERR_READ_USERDATA	= 13 // error while transmitting payload

## EXAMPLE USAGE CHARTS

In order to talk with the module, pull the /CS\_MODULE pin LOW and wait for the module to respond with "RDY\r". After that one can issue a command terminated with '\r' <CARRIAGE RETURN> char. The module will just answer with a simple "ACK\r", depending on the command followed with desired information (e.g. SVN). See Figure 2 example of simple commands.

If a command is not recognized (command not known, typo, wrong parameters, timeout, etc.) the label will respond with an Error followed by the Error code (e.g. ERR 0200 = ERR\_COMMAND\_TIMEOUT.) see Figure 3 error responses

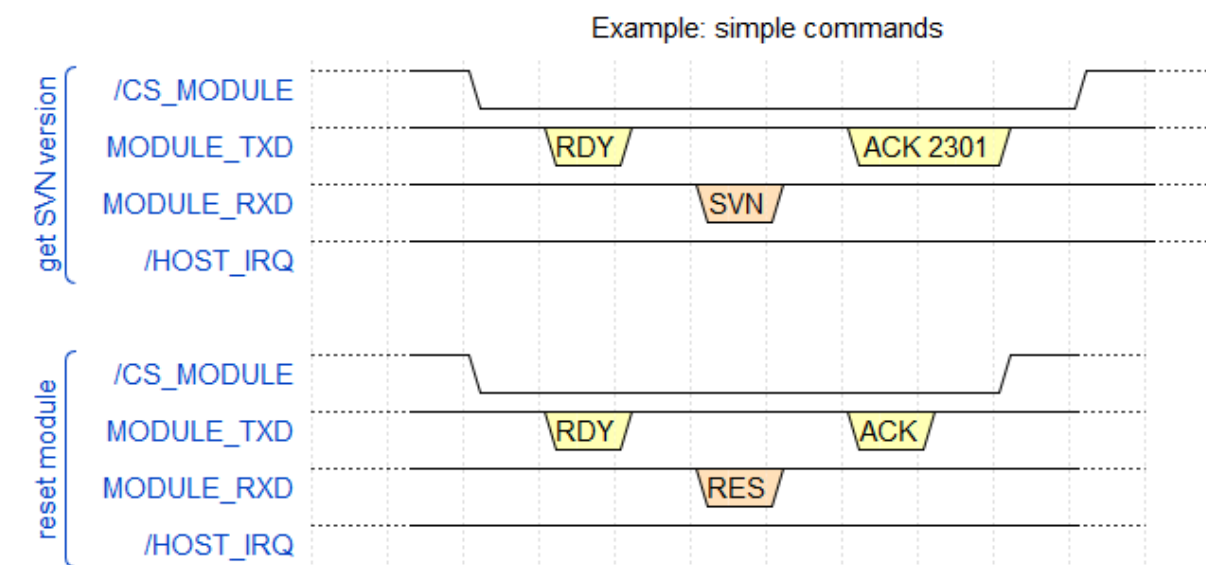


Figure 2 example of simple commands

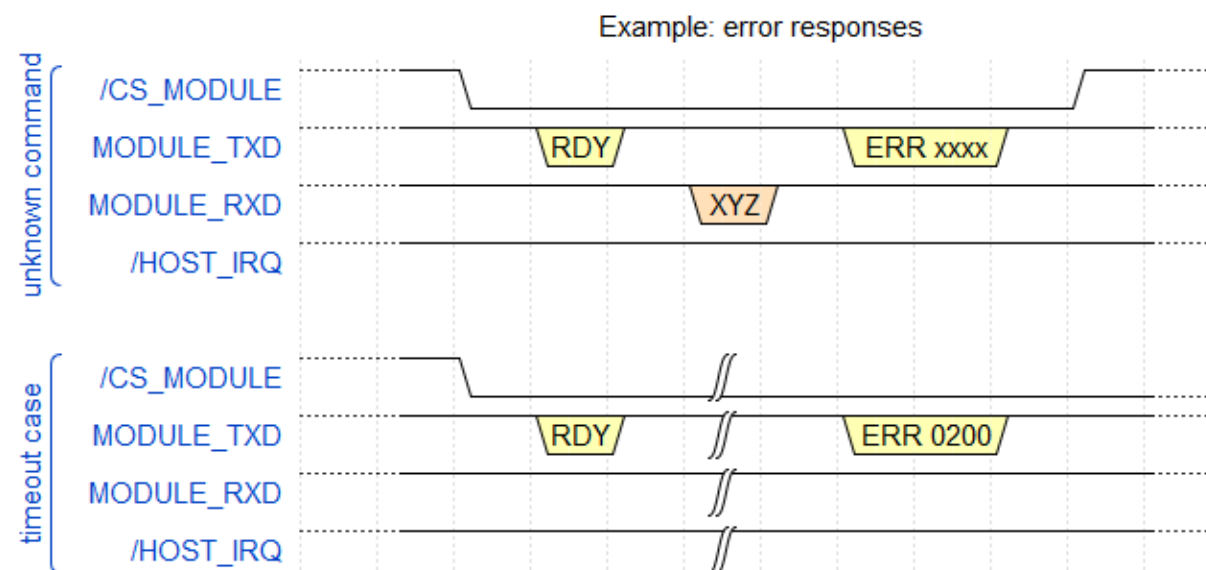


Figure 3 error responses

In order to enable host interrupts, use the command “EHI” followed by the desired interrupts to enable. A tool to provide help on how to select interrupts and the corresponding command is provided (see interrupt\_command.py). see Figure 4 enable host interrupts.

There is no need to enable an interrupt again once it was activated.

To disable the interrupt again, issue an “EHI 00000000\r” command to disable ALL interrupts or select a specific interrupt to disable (again see interrupt\_command.py for help)

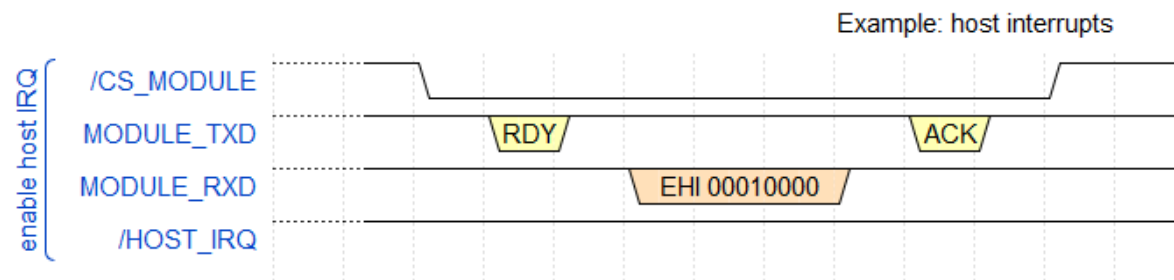


Figure 4 enable host interrupts

After an interrupt was enabled (e.g. set\_config #0 was written) the module will pull the /HOST\_IRQ pin LOW. After that the host can communicate with the module (read set\_config, ...).

It is important to clear the active interrupt flag again to allow the /HOST\_IRQ to go HIGH again.

To do this one can simply clear ALL active interrupts (CHI FFFFFFFF\r) or ask for the currently active interrupts (“GHI\r”) and just clear those. see Figure 5 handle host interrupts and clear active flags.

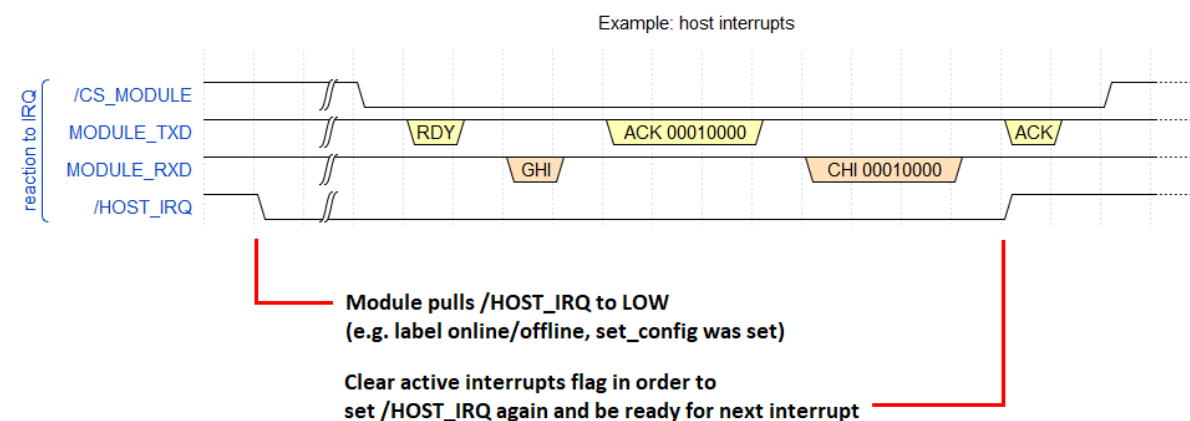


Figure 5 handle host interrupts and clear active flags



## PRODUCT LABELING

---

The Host Marketing Name (HMN) shall be indicated on the exterior of the host product or on the product packaging, or in the product literature, which shall be supplied with the host product or readily available online.

The host product shall be properly labelled to identify the modules within the host product.

The ISED certification label of a module shall be clearly visible at all times when installed in the host product; otherwise, the host product must be labelled to display the ISED certification number for the module, preceded by the word "contains" or similar wording expressing the same meaning, as follows:

Contains IC: 27854-RMG3RCOMA

In addition to containing other required statements specified elsewhere in this standard or in the applicable RSS, user manuals for licence-exempt radio apparatus shall contain the following text, or an equivalent notice, that shall be displayed in a conspicuous location, either in the user manual or on the device, or both in both English and French:

*This device contains licence-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's licence-exempt RSS(s). Operation is subject to the following two conditions:*

*(1) This device may not cause interference.*

*(2) This device must accept any interference, including interference that may cause undesired operation of the device.*

L'émetteur/récepteur exempt de licence contenu dans le présent appareil est conforme aux CNR d'Innovation, Sciences et Développement économique Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

*(1) L'appareil ne doit pas produire de brouillage;*

*(2) L'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.*

## RF EXPOSURE

---

The module is an exempt RF device. It can be used in hosts for portable and mobile applications and for fixed installation. This module can not be used for medical implants.

The module comes with a permanent antenna assembled.

The host integrator is not allowed to change the antenna or any RF impacting parameters.

RF integration specifications:

- Usable band: 2404,053 MHz – 2479,285 MHz
- 0-10 Channels available
- Transmission power of  $\leq 10$  dBm

The host integrator must comply with the following requirements:

- The position of the module in the host system has to be verified and approved by the module provider.
- Every change in the host system (PCB dimensions or mechanical changes) needs to be logged and communicated to the module provider.
- The PCB ground must not be connected directly to the casing material
- The module has to have a minimum distance of 10 mm to the metal casing.
- Furthermore, the module has to have at least 25 mm distance to batteries.
- The module needs to be reachable, meaning it is not allowed to be mounted behind metal or liquids. Instead, the host system cover must be some kind of plastic to cover the opening in the metal in front of the antenna. A typical plastic with a permittivity of  $\sim 2.5$  should be used.
- The usage of the module is permitted only in case the host integrator complies with all RF exposure requirements.

## SPURIOUS EMISSION TESTS

---

For spurious emission tests, the host integrator can flash a dedicated certification firmware. This firmware can

- (a) operate on a defined channel in the 2.4 GHz spectrum
- (b) send permanently data
- (c) receive data and alert package loss

## FCC COMPLIANCE

---

The following rules are applicable to this modular transmitter:

- 1.1307 Actions that may have a significant environmental effect, for which Environmental Assessments (EAs) must be prepared.
- 2.1033 Application for certification.
- 15.209 Radiated emission limits; general requirements.
- 15.249 Operation within the bands 902-928 MHz, 2400-2483.5 MHz, 5725-5875 MHz, and 24.0-24.25 GHz.

This limited modular transmitter is only FCC authorized for the Part 15 FCC transmitter rules listed in the grant. The host product manufacturer is responsible for compliance to any other FCC rules that apply to the host not covered by the limited modular transmitter grant of certification. The host product still requires Part 15 Subpart B compliance testing with the modular transmitter installed.

A host product shall use a physical label stating "Contains Transmitter Module FCC ID: 2A3HY-RMG3-RCOM-A," or "Contains FCC IDs: 2A3HY-RMG3-RCOM-A, XYZMODEL," or shall use e-labeling."

The antenna(s) used for this transmitter must not transmit simultaneously with any other antenna or transmitter, except in accordance with FCC multi-transmitter product procedures.

The device shall bear the following statement in a conspicuous location on the device:

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

When the device is so small or for such use that it is impracticable to label it with the statement specified under paragraph (a) of this section in a font that is four-point or larger, and the device does not have a display that can show electronic labeling, then the information required by this paragraph shall be placed in the user manual and must also either be placed on the device packaging or on a removable label attached to the device.

The users manual or instruction manual for an intentional or unintentional radiator shall caution the user that changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment. In cases where the manual is provided only in a form other than paper, such as on a computer disk or over the Internet, the information required by this section may be included in the manual in that alternative form, provided the user can reasonably be expected to have the capability to access information in that form.

A compliance information statement that includes the following items (Section 2.1077(a)) must be supplied with the product at the time of marketing or importation

- Identification of the product, e.g., trade name, model, etc.
- A statement that the product complies with the rules, as applicable; and
- The name and address, and telephone number, or internet contact information of the responsible party's (as defined in Section 2.909(b)) contact located in the United States.

Further to Section 2.1077(c), the compliance information must be provided in a form that an end-user can reasonably be expected to have the capability to access, such as the instruction manual, a separate product insert, computer disk, web page, or the device's own electronic screen. Compliance information may be provided electronically as permitted in Section 2.935. "

Devices authorized under the SDoC procedure have the option to use the FCC logo to indicate compliance with the FCC rules,<sup>12</sup> and the logo may be included in the instruction materials or as part of an e-label.

The FCC logo shall only be used on a product that has been tested, evaluated, and found to be compliant in accordance with the SDoC procedures. The use of the FCC logo on the device does not mitigate the requirement to provide a means to uniquely identify the product or to provide the required compliance information statement. The FCC logo cannot be used on products that are exempt from an authorization by rule (e.g., Section 15.103 exempt devices, or Section 15.3 incidental radiators) unless the SDoC procedure has been fully applied for the product.