



TI-Rover BT User Guide

Learn more about TI Technology through the online help at education.ti.com/eguide.

Important Information

Except as otherwise expressly stated in the License that accompanies a program, Texas Instruments makes no warranty, either express or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding any programs or book materials and makes such materials available solely on an "as-is" basis. In no event shall Texas Instruments be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of these materials, and the sole and exclusive liability of Texas Instruments, regardless of the form of action, shall not exceed the amount set forth in the license for the program. Moreover, Texas Instruments shall not be liable for any claim of any kind whatsoever against the use of these materials by any other party.

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Texas Instruments is under license.

© 2026 Texas Instruments Incorporated.

Actual products may vary slightly from provided images.

Contents

TI-Rover BT User Guide **1**

 TI-Rover BT 1

 Preparing TI-Rover BT 2

 Pairing TI-Rover BT with a compatible calculator 3

 Charging TI-Rover BT 5

 TI-Rover BT On-Board Components 6

Available commands (in the Python editor) **10**

 Drive 10

 Math Drive 12

 I/O 14

 Commands 16

FAQ's and Troubleshooting **18**

General Information **19**

 Online Help 19

 Contact TI Support 19

 Service and Warranty 19

 License Information 19

General Precautions **20**

 TI Bluetooth® Adapter 20

Regulatory info **20**

 FCC Statement 20

 Canada Declaration 21

TI-Rover BT User Guide

TI-Rover BT

Meet **TI-Rover BT** – the compact, Bluetooth®-enabled robot that lets students code, draw, and explore math when connected wirelessly to their TI-84 Evo.

TI-Rover BT is an educational robotic vehicle so students in math and STEM classrooms can explore graphing, coding, and robotics. TI-Rover BT works with the TI-84 Evo graphing calculator using OS version 7.1 or higher and a TI Bluetooth® Adapter. Students can write python programs on their TI-84 Evo and send the program wirelessly to their TI-Rover BT with the TI Bluetooth® Adapter. The TI-Rover BT can draw shapes, functions, and various designs using the Pen Up/Pen Down system with a fine or ultra fine popular dry-erase marker.

TI-Rover BT can draw several different kinds of mathematical representations such as linear and non-linear functions, sine, cosine, circle, and ellipses... using the “Math Drive” python commands. Precision motors enable TI-Rover BT to navigate complex mazes or draw intricate images on chart paper or dry-erase surfaces. Built-in to TI-Rover BT is an RGB LED, ranger, and two ports for SEEED Studio Grove devices. Use the included USB-C port to charge the built-in Lithium-Ion battery or update the TI-Rover BT with any future firmware updates.

Preparing TI-Rover BT

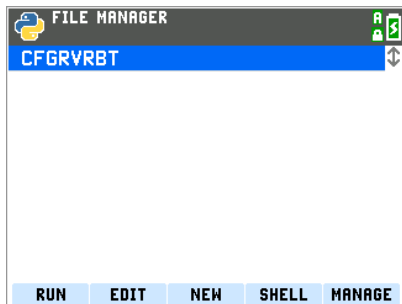
- TI-Rover BT is compatible with TI-84 Evo with OS 7.1 or higher.
- Preparing TI-Rover BT
 - TI-Rover BT comes with all of the necessary equipment to use with a compatible calculator out of the box. You should have (1) TI-Rover BT, (1) TI Bluetooth® Adapter with cable (1) USB A to USB C cable for charging and TI-Rover BT updates, and (1) USB-mini to USB C adapter to make the TI Bluetooth® Adapter compatible with a TI-84 Evo's USB C port.



- Download the Bluetooth® configuration file found at www.education.ti.com/RoverBT. This file is used to pair your TI-Rover BT to the TI Bluetooth® adapter connected to your calculator. The file is titled, **CFGRVRBT**
- To get the file onto your calculator, use the free online connectivity application at connectevo.ti.com and then follow the prompts.

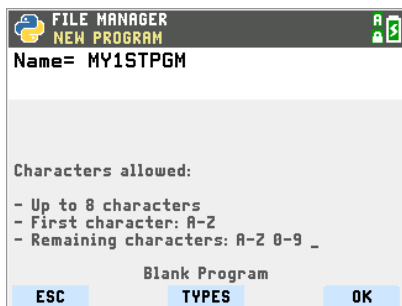
Pairing TI-Rover BT with a compatible calculator

1. Plug the TI Bluetooth® Adapter into the USB port of a compatible calculator.
2. Turn on the power to the calculator and the TI-Rover BT.
3. On your calculator select Python and choose the configuration file. For TI-84 Evo, find the file titled, **CFGRVRBT**.

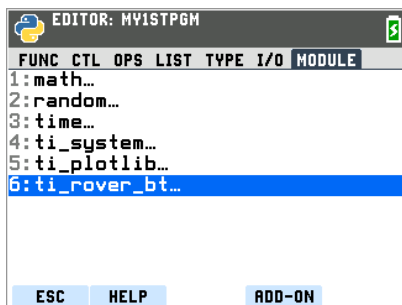


This is the file that will find available TI-Rover BT's and lets you choose which one to associate the connected TI Bluetooth® Adapter to. Run the program, follow the prompts to connect your calculator to an available TI-Rover BT.

4. Once connected, quit the **CFGRVRBT** program. If for some reason the program doesn't respond, you can always quit any program by pressing the **On** key a few times until the program quits.
5. Once you are back to the File Manager screen, you can choose already made programs or create a new program. To create a new program, select **New**, give your program a name, and then select **OK**. That will take you to the Editor where programs are written.



To write your program, select, **FNS...** (functions) and then scroll to **MODULE** and choose, **ti_rover_bt**. Now choose **import ti_rover_bt as rv** as the first line in your program. This is the import statement which enables the calculator to know that it's working with a TI-Rover BT.



6. Now you'll repeat the process to add lines of code to your program. You'll mostly be repeating many of the same steps as above... Choose **FNS...**, go to **MODULE...** select **ti_rover_bt** to find commands and methods for your TI-Rover BT device. Scroll across the top of the menu to see options for Drive, Math Drive, I/O, and Commands.

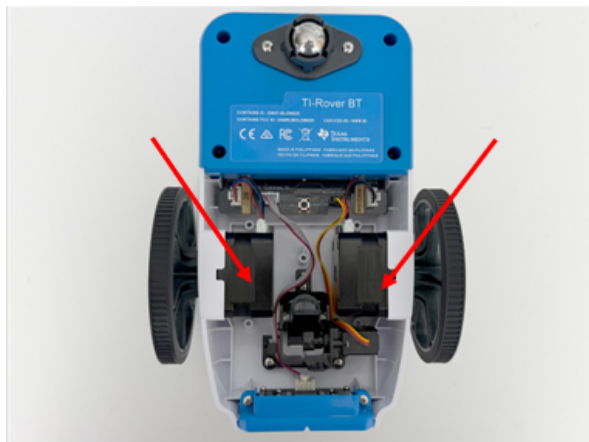
Charging TI-Rover BT

TI-Rover BT comes with a standard A to C USB cable. This cable can be used to charge TI-Rover BT. Charging from mostly depleted to full battery should take a few hours. While TI-Rover BT is charging, the charge LED will appear orange in color. Once charging is complete, the LED will turn green. A full charge should last approximately 1 full day of typical classroom usage.



Ti-Rover BT On-Board Components

- **Wheels/motors** - Each with stepper motors for precise movement



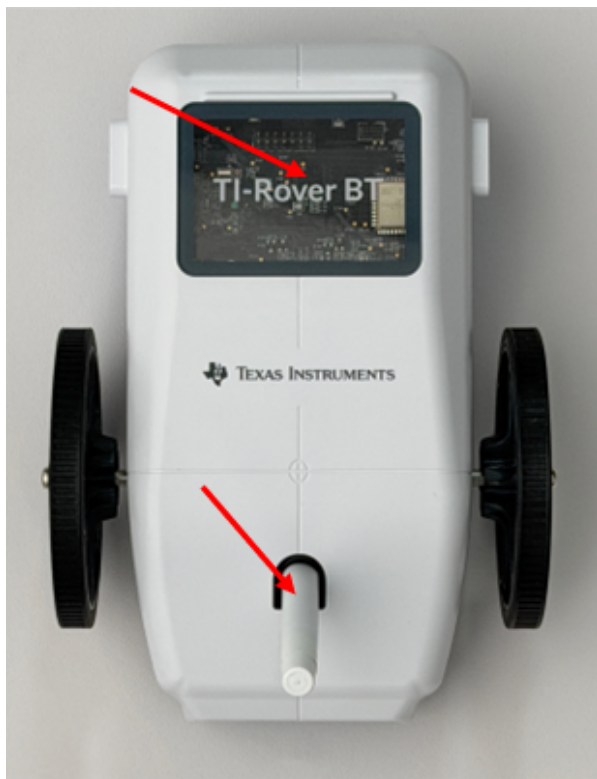
- **Ultrasonic Ranger** - Front-mounted for measuring distance to obstacles



- **On/Off (I/O) Button** - For turning TI-Rover BT on or off
- **Two Grove I/O ports**
- **USB C Port** - For charging and software updates



- **Marker Holder** - Use for holding a dry-erase marker. A fine or ultra fine dry-erase marker is required so that it snaps into place into the pen holder properly.
- **Red-Green-Blue (RGB) LED** - Mix colors through Python coding under the I/O menu.

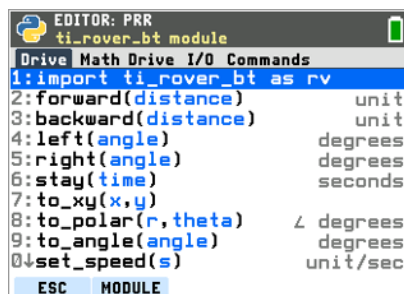


- Attachment points for popular block and stick building systems.



Available commands (in the Python editor)

Drive

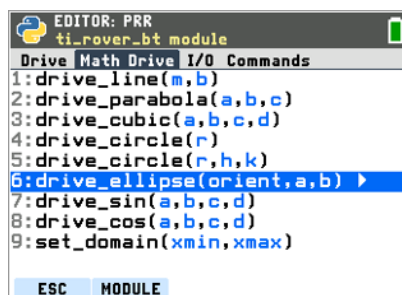


The screenshot shows a Python editor window titled "EDITOR: PRR" with a sub-header "ti_rover_bt module". A menu bar at the top includes "Drive", "Math", "Drive", "I/O", and "Commands". The "Drive" menu is open, displaying a list of commands with their units: `1: import ti_rover_bt as rv`, `2: forward(distance)` (unit), `3: backward(distance)` (unit), `4: left(angle)` (degrees), `5: right(angle)` (degrees), `6: stay(time)` (seconds), `7: to_xy(x,y)`, `8: to_polar(r, theta)` (degrees), `9: to_angle(angle)` (degrees), and `10: set_speed(s)` (unit/sec). At the bottom of the editor, there are buttons for "ESC" and "MODULE".

Drive	Syntax	Description
<code>import ti_rover_bt as rv</code>	<code>import ti_rover_bt as rv</code>	Import statement required to operate TI-Rover BT
<code>forward(distance)</code>	<code>rv.forward()</code>	Rover moves forward a given distance Distance if specified is in centimeters. Default: 1 centimeter Range: -1000 to 1000 A negative number results in a backward motion Note: optionally, you can add a value for speed (max = 100 cm/s) and acceleration (max = 50 cm/s) to the <code>rv.forward</code> or <code>rv.backward</code> (distance value, speed value, acceleration value)
<code>backward(distance)</code>	<code>rv.backward()</code>	Rover moves backward a given distance Distance if specified is in centimeters. Default: 1 centimeter Range: -1000 to 1000 A negative number results in a forward motion
<code>left(angle) degrees</code>	<code>rv.left()</code>	Turn Rover to the left at specified angle. The turn will execute as a spin

		motion. Default turn is 90 degrees. Range: -1800 to 1800 Negative value means a 'right' turn
right(angle) degrees	rv.right()	Turn Rover to the right at specified angle. The turn will execute as a spin motion. Default turn is 90 degrees. Range: -1800 to 1800 Negative value means a 'left' turn
stay(time) seconds	rv.stay()	Rover stays in place for an optionally specified time. Default time is 30.0 seconds. Range: (0, 100) The motors should stay engaged during the stay() so there's no unexpected movement for the next move command.
to_xy(x,y)	rv.to_xy(,)	Moves Rover from current grid location to the specified grid location (x,y).
to_polar(r,theta) degrees	rv.to_polar(,)	Moves Rover from current grid location to the specified grid location (r,theta). Radius r is in default grid units. Angle theta is in default degrees.
to_angle(angle) degrees	rv.to_angle()	Spins Rover to the specified angle from current heading.
set_speed (unit/sec)	rv.set_speed()	Changes the default speed (40 cm/sec) Range: (1,100) cm/sec
set_acceleration (unit/sec ²) or set_accel	rv.set_acceleration()	Changes the default acceleration (10 cm/sec/sec) Range: (0,50) cm/sec²
forward_time (time)	rv.forward_time()	Rover moves forward at default speed for given time
backward_time (time)	rv.backward_time()	Rover moves backward at default speed for given time

Math Drive



```

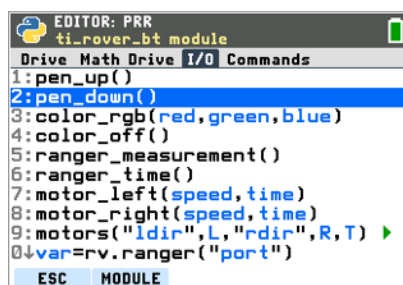
EDITOR: PRR
ti_rover_bt module
Drive Math Drive I/O Commands
1: drive_line(m,b)
2: drive_parabola(a,b,c)
3: drive_cubic(a,b,c,d)
4: drive_circle(r)
5: drive_circle(r,h,k)
6: drive_ellipse(orient,a,b) ▶
7: drive_sin(a,b,c,d)
8: drive_cos(a,b,c,d)
9: set_domain(xmin,xmax)

ESC  MODULE
  
```

Math Drive	Syntax	Description
drive_line(m,b)	# $y = mx + b$ rv.drive_line(,)	indicates the cursor position - at the location for the first argument
drive_parabola(a,b,c)	# $y = ax^2 + bx + c$ rv.drive_parabola(, ,)	indicates the cursor position - at the location for the first argument
drive_cubic(a,b,c,d)	# $y = ax^3 + bx^2 + cx + d$ rv.drive_cubic(, , ,)	indicates the cursor position - at the location for the first argument
drive_circle(r)	# $r = \text{radius}$ rv.drive_circle()	$r = \text{radius}$ - starting point in the bottom of the circle indicates the cursor position - at the location for the first argument
drive_circle(r,h,k)	# $(x-h)^2 + (y-k)^2 = r^2$ rv.drive_circle(, ,)	indicates the cursor position - at the location for the first argument RoverBT will start the movement/drawing from the bottom of the circle
drive_ellipse(orientation, a, b)	# $a = \text{major axis}, b = \text{minor axis}$ rv.drive_ellipse(orient,,)	$a = \text{major axis}, b = \text{minor axis}$ orientation - Options: horizontal, vertical RoverBT will start the movement/drawing from the bottom of the ellipse
drive_sin(a,b,c,d)	# $a \cdot \sin(b(x-c)) + d$ rv.drive_sin(, , ,)	$a \cdot \sin(b(x-c)) + d$ indicates the cursor position -

		at the location for the first argument Note: Values that cause extremely acute movements will likely result in an error
<code>drive_cos(a,b,c,d)</code>	# $a \cdot \cos(b(x-c)) + d$ <code>rv.drive_cos(, , ,)</code>	$a \cdot \cos(b(x-c)) + d$ indicates the cursor position - at the location for the first argument Note: Values that cause extremely acute movements will likely result in an error
<code>set_domain(xmin, xmax)</code>	# default (0,10) <code>rv.set_domain(,)</code>	default (0,10) indicates the cursor position - at the location for the first argument This applies to line, parabola, sin and cos functions xmin range: (-20, 20) xmax range: (-20, 20) Values outside this range will trigger an exception

I/O



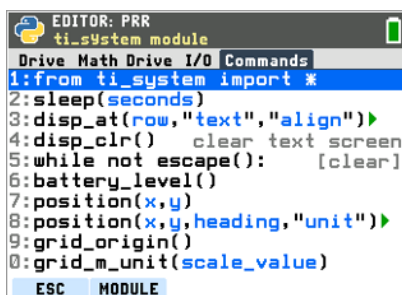
```

EDITOR: PRR
ti_rover_bt module
Drive Math Drive I/O Commands
1: pen_up()
2: pen_down()
3: color_rgb(red,green,blue)
4: color_off()
5: ranger_measurement()
6: ranger_time()
7: motor_left(speed,time)
8: motor_right(speed,time)
9: motors("ldir",L,"rdir",R,T)
10: var=rv.ranger("port")
ESC MODULE
  
```

I/O	Syntax	Description
pen_up()	rv.pen_up()	Moves the pen up. If the pen is already up, it will stay up.
pen_down()	rv.pen_down()	Sets the pen down. If pen is already down, it will stay down.
color_rgb (red,green,blue)	#rv.color_rgb (red,green,blue) rv.color_rgb(l,,)	Sets the RGB LED output to the specified r,g,b values, with each value between 0 to 255.
color_off()	rv.color_off()	Turns off the RGB LED
ranger_ measurement()	rv.ranger_ measurement()	Reads the ultrasonic distance sensor on the front, returning the current distance in meters. If no reading is possible (ex. object is too far away), a value of 10 meters is returned. Returned values are rounded to 3 decimal places.
ranger_time()	rv.ranger_time()	Returns the time of flight out and back for the ultrasonic echo. Time is returned in seconds. If no reading is possible, returned value is 10 seconds.
motor_left (speed,time)	#rv.motor_left (speed,time) rv.motor_left(l,)	Speed is specified in "cm per second".
motor_right (speed,time)	#rv.motor_right (speed,time) rv.motor_right(l,)	Speed is specified in "cm per second".

motors("ldir", left_val, "rdir", right_val, time)	#rv.motors ("ldir",L,"rdir",R,T)> clockwise; counterclockwise	ldir(left direction), rdir(right direction) is specified thorough an argument helper with "cw" (clockwise) and "ccw" (counter clockwise) as the options. Speed is specified in "cm per second".
Ranger_ext		Argument helper options:
var=rv.ranger ("port")	=rv.ranger(" ")	Port 1, Port 2
var.measurement ()	.measurement()	
var.measurement_ time()	.measurement_ time()	
Digital In Out		Argument helper options:
var=rv.digital ("port")	=rv.digital ("port#")	Port1, Port2
var=rv.led("port")	=rv.led("")	
var.measurement ()	.measurement()	
var.set(val)	.set()	set: range (0,1)
var.get()	.get()	get: returns 0 or 1
var.off()	.off()	
var.on()	.on()	
var.blink (freq,time)	.blink(,)	blink: should be in Hz and stop after time in seconds
var.toggle (freq,time)	.toggle(,)	toggle: should be in Hz and stop after time in seconds
var.pwm (duty,freq,time)	.pwm(,,)	pwm: should be in Hz and duty is in percent and should stop after time in seconds

Commands



```

EDITOR: PRR
ti_system module
Drive Math Drive I/O Commands
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align")
4:disp_clr()      clear text screen
5:while not escape():  [clear]
6:battery_level()
7:position(x,y)
8:position(x,y,heading,"unit")
9:grid_origin()
0:grid_m_unit(scale_value)
ESC  MODULE
  
```

Commands	Syntax	Description
from ti_system import *	from ti_system import *	This comes from the ti_system module menu and will use the same strings from there
sleep(seconds)	sleep(seconds)	This comes from the ti_system module menu and will use the same strings from there
disp_at(row, "text", "align")	disp_at(l,"","")	This comes from the ti_system module menu and will use the same strings from there
disp_clr()	disp_clr()	This comes from the ti_system module menu and will use the same strings from there
while not escape():	while not escape():	This comes from the ti_system module menu and will use the same strings from there
battery_level()	var = rv.battery_level()	Returns a number between 0 and 100 [Clear] key will break the loop
position(x,y)	rv.position(,.)	Identifies Rovers current x and y coordinate position
position (x,y,heading,"unit")	# rv.position (x,y,heading,"unit") rv.position(,,,"")	"unit" will be the existing argument helper with 1. degrees 2. radians 3. grads as options
grid_origin()	rv.grid_origin()	Sets the virtual grid origin of the TI-

		Rover to (0,0) without moving the TI-Rover.
grid_m_unit(scale_value)	rv.grid_m_unit()	Sets the virtual grid spacing in meters per unit (m/unit) to the value given. 0.01 is the default m/unit and translates to 1 unit = 100 mm or 10 cm or 1 dm or 0.1 m. The range of valid scale_value is from 0.01 to 10.0.

If you'd like to learn more about the Python Editor and programming options beyond TI-Rover BT go to <https://education.ti.com/en/product-resources/eguides/eguide-84-evo/python>

FAQ's and Troubleshooting

The marker is not making contact with the surface even when the pen down command is active -

Ensure the marker is completely snapped into the marker holder. This involves using a little bit of force to push a compatible marker down into the holder until a loud snap is heard. Ensure you are using a compatible marker. The popular brand of fine and ultra fine dry-erase markers are supported. If the marker is supported and fully snapped into the marker holder and it is still not making contact with the intended surface, consider gently gripping the tip with a pair of pliers and pulling it out of the barrel very carefully and slightly to lengthen the tip.

The TI Bluetooth® Adapter is not making contact with the TI-Rover BT

Make sure power is on for both the calculator you are using that has the TI Bluetooth® Adapter connected to it and the TI-Rover BT. If power is on for both, you should see a white flashing LED light on the TI-Rover BT and a solid white light on the TI Bluetooth® Adapter. If one or the other device has a colored light it means that device is already paired and will need to be disconnected. Once properly paired the LED lights on each device should be the same color indicating they are paired with one another. If either device is still showing a white LED light and they still won't pair with one another, you may need to update your TI Bluetooth® Adapter's firmware. You can do this with a supported Windows-based computer and TI's free downloadable application found here - education.ti.com/bluetooth

How long does the battery in TI-Rover BT last?

The battery should last one full school day of periodic use before needing to be recharged.

TI-Rover BT's python module has a menu called, "Math Drive". What does it do?

Math drive is a unique feature that enables students to program their TI_Rover BT to draw various linear and non-linear functions as well as circles and ellipses with high precision. More information may be available at education.ti.com/roverbt

General Information

Online Help

education.ti.com/eguide

Select your country for more product information.

Contact TI Support

education.ti.com/ti-cares

Select your country for technical and other support resources.

Service and Warranty

education.ti.com/warranty

Select your country for information about the length and terms of the warranty or about product service.

Limited Warranty. This warranty does not affect your statutory rights.

License Information

education.ti.com/license

Select your country for more license information.

General Precautions

TI Bluetooth® Adapter

- Do not expose the TI Bluetooth® Adapter to temperatures above 140°F (60°C).
- Use only the USB cable provided with the TI Bluetooth® Adapter when connecting to a TI graphing calculator.

Additional Information:

- Operating Temperature: 0°C to 45°C
- Operating Voltage: 5Vdc
- Operating Current: 20mA

Regulatory info

- FCC Statement
- Canada Declaration

FCC Statement

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.

NOTE The manufacturer is not responsible for any radio or TV interference caused by unauthorized modifications or changes to this equipment. Such modifications or changes could void the user's authority to operate the equipment.

NOTE This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: - Reorient or relocate the receiving antenna. - Increase the separation between the equipment and receiver. - Connect the equipment into an outlet on a circuit different from that to which the receiver is connected. - Consult the dealer or an experienced radio/TV technician for help.

Canada Declaration

Canada IC RSS-GEN Issue 5 section 4.3 – “Contains transmitter module IC: 25657-BLEM205”

This device contains license-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada’s license-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.

CAN ICES-3(B)/NMB-3(B)