

TI-Nspire™ CX II – CoDrone EDU Getting Started Guide

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.

© 2025 Texas Instruments Incorporated

"Python" and the Python logos are trademarks or registered trademarks of the Python Software Foundation, used by Texas Instruments Incorporated with permission from the Foundation.

ROBOLINK 

Robolink makes CoDrone EDU, the drone of choice for classrooms and the REC Foundation's Aerial Drone Competition. Designed for grades 5 and up, CoDrone EDU helps students build STEM, coding, and future-ready skills from their first flight. Learn more at www.robolink.com.

Actual products may vary slightly from provided images.

Contents

- Getting Started** **1**
 - Preparing the CoDrone EDU System 1
 - Writing Your First Flight Program 3

- CoDrone EDU Python Function Reference** **7**
 - General 7
 - Flight Start/Stop 7
 - Flight Movements 8
 - Flight Controls 10
 - Flight Stunts 11
 - LED and Buzzer 12
 - Sensors 12
 - Controller Screen 14
 - Commands 14

- Things to Know** **16**

- Troubleshooting** **17**

- General Information** **18**

Getting Started

Preparing the CoDrone EDU System

To control a CoDrone EDU with a Python program on the TI-Nspire™ CX II, you will need:

- TI-Nspire™ CX II calculator with OS version 6.3 or later
- CoDrone EDU Python module (CoDroneEDU.tns)
- CoDrone EDU (controller and drone)
 - AA batteries for the CoDrone EDU Controller
 - Charged batteries for the CoDrone EDU
- Cable for TI Nspire CX II x CoDrone EDU ([sold separately](#))

Note: USB cables from other vendors will not work properly and should not be used.



Updating the TI-Nspire™ CX II calculator OS

Visit the [TI-Nspire™ CX Connect web tool](#) to update the TI-Nspire™ CX II OS to version 6.3.0 or later.

Note: Perform this step prior to transferring and installing the CoDroneEDU module.

Installing the CoDrone EDU Python module

1. Visit the [TI CoDrone web page](#) to download the module.
2. Use the [TI-Nspire™ CX Connect web tool](#) to transfer the CoDroneEDU.tns file to the calculator.
3. Open the CoDroneEDU.tns file on your calculator and follow the directions on the first page to install the module.

Once the module is installed, the CoDroneEDU.tns will be closed. This is a one-time process and will not need to be repeated.

CoDrone EDU setup

Refer to Robolink's [CoDrone EDU User Manual](#) for more information about the drone before you start flying.

The manual covers:

- What's Included
- Before You Fly
- Getting to Know Your CoDrone EDU
- Getting to Know Your Controller
- Powering On
- Charging
- Pairing
- Basic Flight Commands
- Propellers
- Motors



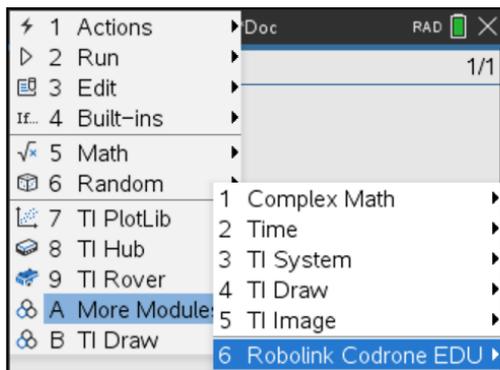
There are also instructional videos and other resources on Robolink's [Basecamp for the CoDrone EDU](#).



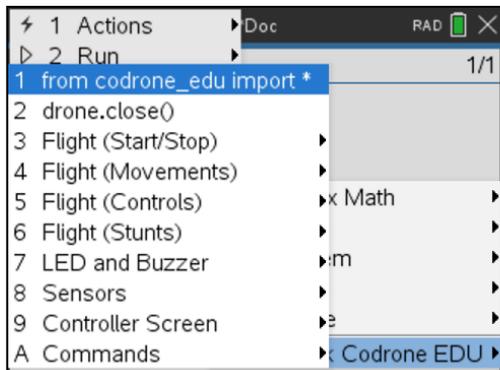
Writing Your First Flight Program

Writing the program

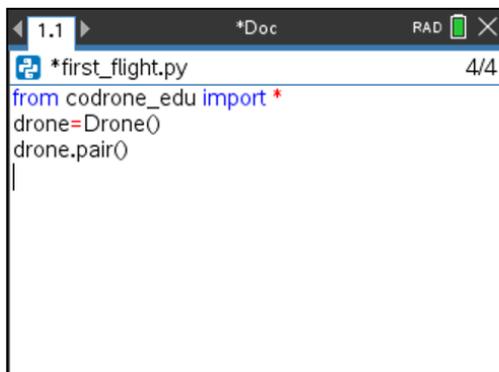
1. Press **menu** and select **Add Python > New**.
2. Enter **First_Flight** in the **Name** field and click **OK**.
3. Press **menu** and select **More Modules > Robolink CoDrone EDU**.



4. Select **from codrone_edu import ***.

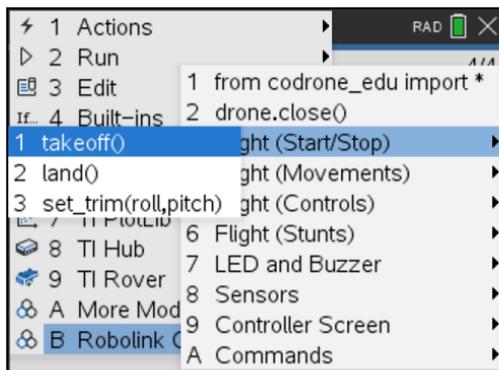


This is required to utilize all of the CoDrone EDU commands and pastes the necessary statements to communicate with the drone.

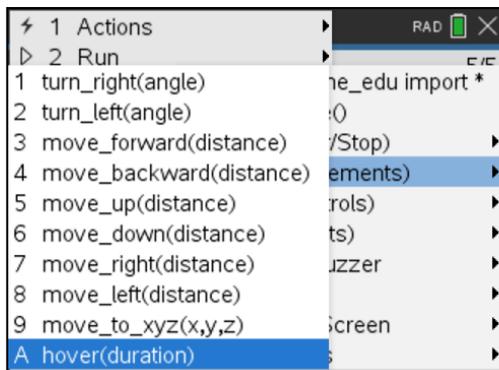


```
1.1 *Doc RAD X
*first_flight.py 4/4
from codrone_edu import *
drone=Drone()
drone.pair()
|
```

5. Press **menu** and select **Robolink CoDrone EDU > Flight (Start/Stop) > takeoff()**.

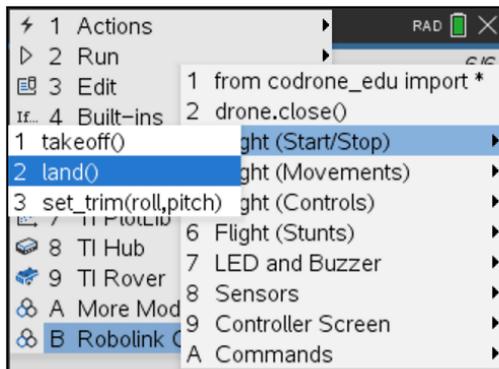


6. Press **menu** and select **Robolink CoDrone EDU > Flight (Movements) > hover(duration)** and enter a small number for the time the drone will hover in seconds.



```
1.1 *Doc RAD X
*first_flight.py 5/5
from codrone_edu import *
drone=Drone()
drone.pair()
drone.takeoff()
drone.hover(seconds)
```

7. Press **menu** and select **Robolink CoDrone EDU > Flight (Start/Stop) > land()**.



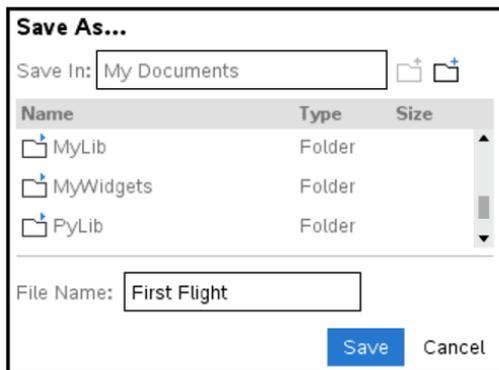
8. Press **menu** and select **Robolink CoDrone EDU > drone.close()**.

The full program will look like this:

```
1.1 *Doc RAD X
*first_flight.py 8/8
from codrone_edu import *
drone=Drone()
drone.pair()
drone.takeoff()
drone.hover(3)
drone.land()
drone.close()
|
```

9. Press **menu** and select **Run > Check Syntax & Save (Ctrl+B)** to save the program.

10. Press **doc** and select **File > Save (Ctrl+S)** to save the TNS document.
11. On the **Save As...** screen, select a folder to save the document to, enter a file name, and click **Save**.



Running the program

1. Connect the CoDrone EDU controller to the TI-Nspire™ CX II calculator using the USB cable.

Note: Only use the TI Nspire CX II x CoDrone EDU cable that is available for purchase on the [Robolink website](#). Do not use cables from other vendors.

2. Place your drone in a well-lighted area free of obstructions.

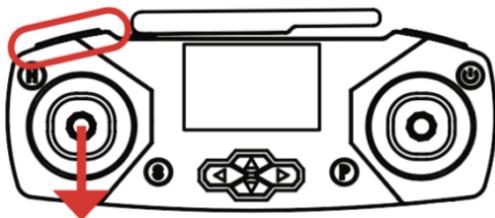
The drone works best when it flies over a floor/surface with a random pattern (wood grain, etc.).

3. Ensure everything is powered up.
4. Press **menu** and select **Run > Run (Ctrl+R)**.

Performing an Emergency Stop

You should be familiar with the Emergency Stop process in case the drone program needs to be halted immediately. **This action will cause the drone motors to stop which will result in the drone falling without any control.** Please exercise caution when using this action.

To perform an Emergency Stop, hold the L1 button on the CoDrone EDU controller while simultaneously pulling down the left joystick to shut the motors off immediately.



CoDrone EDU Python Function Reference

These are the functions and menus in the CoDrone EDU Python module.

General

Menu Selection	Example/Notes
<code>from codrone_edu import *</code>	This module is required to program the drone in Python. Note that when the import statement is pasted into the Python editor, <code>drone = Drone()</code> and <code>drone.pair()</code> are included.
<code>drone = Drone()</code>	Constructs an instance of the Drone class.
<code>drone.pair()</code>	Establishes a radio link between the drone and the controller.
<code>drone.close()</code>	Closes the communication between the TI-Nspire™ CX II and the controller. This command should be included as a best practice at the end of each CoDrone EDU program.
	Example: <code>from codrone_edu import *</code> <code>drone = Drone</code> <code>drone.pair()</code> <code>drone.close()</code>

Flight Start/Stop

Menu Selection	Example/Notes
<code>takeoff()</code>	<code>drone.takeoff()</code> Launches the drone and returns control to the program after the drone stabilizes.
<code>land()</code>	<code>drone.land()</code> Lands the drone and returns control to the program after the drone stabilizes.
<code>set_trim(roll,pitch)</code>	<code>drone.set_trim(-5,0)</code> Sets the roll and pitch trim of the drone in case your drone is drifting.

Menu Selection	Example/Notes
	<p>For example, if the drone is drifting to its right, you may want to set the roll to a small negative value. This trim will remain saved, even after powering off until you have changed the trim either programmatically.</p> <p>Note: If you are setting the trim right before a takeoff, make sure to add a time.sleep(1) before the takeoff(), otherwise the takeoff command might be skipped.</p>

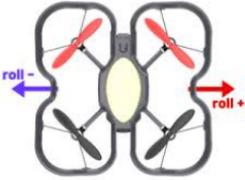
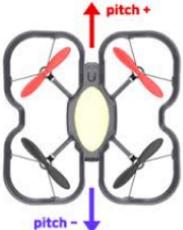
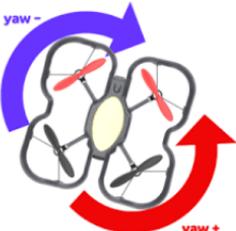
Flight Movements

Menu Selection	Example/Notes
turn_right(angle)	<pre>drone.turn_right(90)</pre> <p>Turns 90° to the right of the drone's forward-facing position.</p>
turn_left(angle)	<pre>drone.turn_left(45)</pre> <p>Turns 45° to the left of the drone's forward-facing position.</p>
move_forward(distance)	<pre>drone.move_forward(50)</pre> <p>Moves forward 50cm from the drone's present position.</p>
move_backward(distance)	<pre>drone.move_backward(150)</pre> <p>Moves backward 150cm from the drone's present position.</p>
move_up(distance)	<pre>drone.move_up(100)</pre> <p>Moves up 100cm from the drone's present position.</p>
move_down(distance)	<pre>drone.move_down(50)</pre> <p>Moves down 50cm from the drone's present position.</p>
move_right(distance)	<pre>drone.move_right(75)</pre> <p>Moves to the right 75cm from the drone's present position.</p>

Menu Selection	Example/Notes
<code>move_left(distance)</code>	<code>drone.move_left(150)</code> Moves to the left 150cm from the drone's present position.
<code>move_to_xyz()</code>	<code>drone.move_to(50,-50,100)</code> Moves the drone to a new position within the absolute 3-D coordinate system, with (0,0,0) as the takeoff point. A positive x-value is forward, a positive y-value is to the left, and a positive z-value is up. Units are centimeters (cm).
<code>hover(duration)</code>	<code>drone.hover(3)</code> Stay at the present position for three seconds.

Flight Controls

Caution! Setting any of the control's power to greater than 50% will cause the drone to move very fast, which may be dangerous in the classroom.

Menu Selection	Example/Notes
<p>set_roll(power)</p> 	<p><code>drone.set_roll(30)</code></p> <p>Set the roll power to 30% to move to the right. Range is -100 to 100.</p>
<p>set_pitch(power)</p> 	<p><code>drone.set_pitch(-10)</code></p> <p>Set the pitch power to -10% to move backward. Range is -100 to 100.</p>
<p>set_yaw(power)</p> 	<p><code>drone.set_yaw(-20)</code></p> <p>Set the yaw power to -20% to turn to the right. Range is -100 to 100.</p>
<p>set_throttle(power)</p>	<p><code>drone.set_throttle(50)</code></p> <p>Set the throttle power to 50% to move up. Range is -100 to 100.</p>

Menu Selection	Example/Notes
	
<code>move(duration)</code>	<code>drone.move(2)</code> Moves the drone for two seconds using roll, pitch, yaw, and throttle flight control settings.

Flight Stunts

Caution! Performing stunts requires significant space to perform the maneuver. Ensure classroom space is cleared before performing these stunts. Flips require a battery charge greater than 50%, and flight control may be erratic after the flip.

Menu Selection	Example/Notes
<code>flip("front")</code>	<code>drone.flip("front")</code> The drone flips forward.
<code>flip("back")</code>	<code>drone.flip("back")</code> The drone flips backward.
<code>flip("right")</code>	<code>drone.flip("right")</code> The drone flips right.
<code>flip("left")</code>	<code>drone.flip("left")</code> The drone flips left.

LED and Buzzer

Menu Selection	Example/Notes
<code>set_drone_LED(R,G,B)</code>	<code>drone.set_drone_LED(255,0,0)</code> Turn on the drone's LED to red.
<code>drone_LED_off()</code>	<code>drone.drone.LED_off()</code> Turn off the drone's LED.
<code>set_controller_LED(R,G,B)</code>	<code>drone.set_controller_LED(0,255,255)</code> Turn on the controller's LED to cyan.
<code>controller_LED_off()</code>	<code>drone.controller_LED_off()</code> Turn off the controller's LED.
<code>drone_buzzer(note,duration)</code>	<code>drone.drone_buzzer("C4",500)</code> Play the note C in the fourth octave (middle C) for 500 milliseconds.
<code>drone_buzzer(frequency,duration)</code>	<code>drone.drone_buzzer(262,500)</code> Play the note C (262Hz) in the fourth octave (middle C) for 500 milliseconds.
<code>controller_buzzer(note,duration)</code>	<code>drone.controller_buzzer("A4",1500)</code> Play the note A in the fourth octave (tuning pitch) for one and a half seconds.
<code>controller_buzzer(frequency,duration)</code>	<code>drone.controller_buzzer(440,1500)</code> Play the note A (440Hz) in the fourth octave (tuning pitch) for one and a half seconds.

Sensors

Menu Selection	Example/Notes
<code>variable = get_battery()</code>	<code>charge = drone.get_battery()</code> Returns the percent charge of the drone's battery.
<code>variable = get_height()</code>	<code>height = drone.get_height()</code> Returns the height of the drone in centimeters.

Menu Selection	Example/Notes
<code>variable = get_front_range()</code>	<pre>x = drone.get_front_range()</pre> <p>Returns the distance to an object in front of the drone in centimeters.</p>
<code>variable = get_heading()</code>	<pre>angle = drone.get_heading()</pre> <p>Returns the drone's heading from the takeoff position (0°).</p>
<code>variable = get_pressure()</code>	<pre>P = drone.get_pressure()</pre> <p>Returns the atmospheric pressure in pascals.</p>
<code>variable = get_temperature()</code>	<pre>T = drone.get_temperature()</pre> <p>Returns the drone's temperature in centigrade degrees.</p>
<code>variable = get_colors()</code>	<pre>color = drone.get_colors()</pre> <p>Returns a list of two elements with the colors detected by the front and the rear color sensors.</p> <p>The color values are: Red, Green, Yellow, Blue, Cyan, Magenta, Black, White, Unknown.</p> <p>Note: The color sensor cannot be used while the drone is in flight.</p>
<code>roll, pitch = get_trim()</code>	<pre>roll, pitch = get_trim()</pre> <p>Returns the current trim values – roll and pitch – of the drone.</p>
<code>time,x,y,z = get_position_data()</code>	<pre>time,x,y,z = get_position_data()</pre> <p>Returns a list of position data for the drone.</p> <p>Note: The drone's position values are reset to 0 when the drone takes off or the battery is re-inserted.</p> <p>Time is the current time of the current program.</p> <p>The 'x' position of the drone is forward and reverse. The 'y' position of the drone is left and right. The 'z' position of the drone is up and down.</p>

Controller Screen

Menu Selection	Example/Notes
<code>controller_draw_string(x,y,"text")</code>	<code>drone.controller_draw_string(10,5,"Howdy")</code> Prints "Howdy" at 10,5 on the controller's LCD screen.
<code>controller_draw_line(x1,y1,x2,y2)</code>	<code>drone.controller_draw_line(0,0,150,75)</code> Draws a diagonal line from (0,0) (in the upper left corner) to (150, 75) (the bottom right corner) on the controller's LCD screen.
<code>controller_clear_screen()</code>	<code>controller_clear_screen()</code> Clears the screen of the controller.

Commands

Menu Selection	Example/Notes
<code>time.sleep(seconds)</code>	<code>time.sleep(2)</code> Stops program execution for two seconds.
<code>while get_key()!="esc":</code>	<pre>the_heights = [] the_times = [] time = 0 while get_key()!="esc": height = drone.get_height() the_heights.append(height) the_times.append(time) time.sleep(1) time += 1</pre> <p>This is an indefinite loop that stores the drone's height in a list named the_heights every second until the calculator's <code>[esc]</code> key is pressed.</p>
<code>store_list("name",list)</code>	<pre>store_list("height",the_heights) store_list("time",the_times)</pre> <p>This saves the Python list named the_heights to the TI-Nspire™ CX II TNS document with the name height and the_times to time. The TI-Nspire™ variables named time and height may be selected in the TI-Nspire™ CX II Data and Statistics</p>

Menu Selection	Example/Notes
	application to make a scatter plot of height vs. time.
clear_history()	clear_history() Clear any previous output in the Python shell when the program containing the function is run.
set_feedback(show/hide)	drone.set_feedback("hide") Controls whether a status feedback is printed after each command. Default setting is show .
module version()	module_version() Returns the version of the CoDrone EDU module installed on the calculator. This helps maintain the latest version of the module. Updates to the module will be posted here .

Things to Know

- When the controller batteries get low, unplugging the CoDrone EDU controller from a TI-Nspire™ CX II and re-plugging into the same or different TI-Nspire™ CX II sometimes freezes the controller. If this happens, try removing the batteries and re-installing them. If the controller is still frozen, try replacing the batteries with new ones.
- For best results, please make sure the drone is flying over a well-lighted floor/surface with a random pattern (wood grain, etc.). ([source](#))
- If you want to take off and immediately land, it is recommended to run a **hover()** or **time.sleep()** command in between the **takeoff()** and **land()** commands. Otherwise the drone may miss the **land()** command and remain in flight. ([source](#))
- The drone's internal coordinate system resets to (0, 0, 0) every time you insert the battery, and again each time you take off. If you try to read or print sensor data while walking the drone to its takeoff location, it may be inaccurate. ([source](#))
- The drone takes three to four seconds after a flip before it can perform another flight command. ([source](#))
- The color sensor only works when the drone has landed and is NOT flying. ([source](#))

Troubleshooting

Problem	Solution
I don't see the CoDrone EDU module menus.	Transfer and install the CoDrone EDU module to the calculator. See Part 1 – Loading the Software.
The calculator will not connect to the drone.	<ul style="list-style-type: none">• Unplug the controller from the calculator and re-connect it.• If the problem persists, remove the AA batteries from the controller and re-install them.
Command status reports "COMPLETED," but the drone flight is erratic.	The drone uses a flow sensor for positioning. A well-lighted space and a floor/surface with a random pattern (wood grain, etc.) are required. A typical glossy-white school flooring may cause erratic flight. Try moving to an area with a different floor appearance with lots of light.
The drone will not take off, the status light is blinking red, and there is an audible beep.	Charge and replace the drone battery.
The drone will not perform flips.	The battery charge must be greater than 50%.
The drone does not respond/will not link to the calculator.	Make sure you are using the USB cable specifically made for the TI-Nspire™ CX II and CoDrone EDU sold on the Robolink website .

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.

Contact Robolink Support

For help with the CoDrone EDU, please visit [this website](#) or send an email to support@roboink.com.

Service and Warranty Information

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.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243