

## TI-84 PLUS CE PYTHON

### Getting Started with Tello

Hello Tello

#### Intro 1

These mini-projects are designed to help you learn to operate the Tello flying drone using your TI-84 Plus CE with Python programming.

They assume you have a working knowledge of Python programming on the TI-84 Plus CE Python edition. While the hardware setup requires a micro:bit, expansion board, and wi-fi adapter, the Tello functions and methods do not require coding skills involving these intermediate devices.



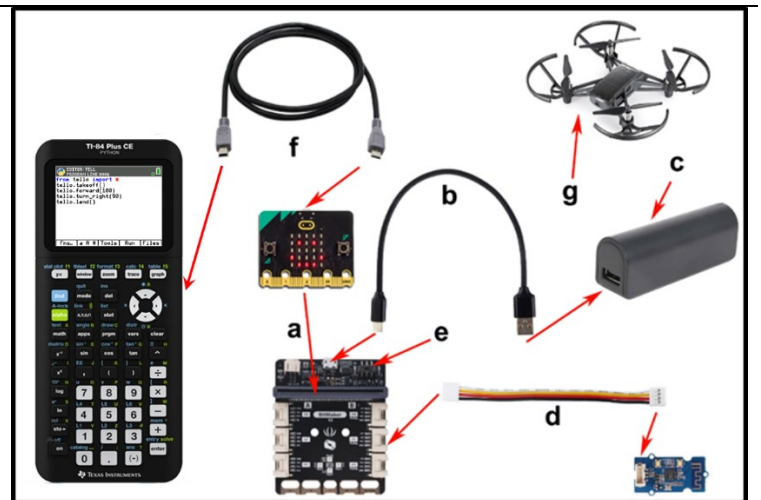
#### Intro 2

The TI-84 Plus CE Python communicates with the Tello using:

- the Tello Python module,
- a BBC micro:bit v2 board with the appropriate TI runtime,
- a micro:bit expansion board,
- a small wi-fi board,
- an external power supply.

You can watch the [YouTube Videos](#) for an overview of this process or read the setup instructions included in the download package. The necessary files (tello.8xv, tellocfg.8xv, and ti\_runtime\*.hex) along with detailed step-by-step setup instructions are available [here](#).

This first mini-project just gets the Tello drone off the ground...



The setup and parts seen here involves connecting the hardware, installing the Tello Python module to your TI-84, installing the special TI-developed micro:bit .hex file, and then configuring the system (pairing the Tello drone with the micro:bit).

## TI-84 PLUS CE PYTHON

### Intro 3

Tello needs to 'see' the ground and a colorful, patterned surface helps Tello maintain its position in the air. Tello has a camera on the bottom and a 'Vision Positioning System' to help it stay in one place while hovering. If the surface below Tello is a plain, solid color (that is, no pattern) then Tello switches to an 'Attitude Positioning System' which is not as reliable as its Vision System so be careful. The blinking LED on the front of Tello indicates which system is in use (green=Vision, red= Attitude). This is explained in more detail in the Tello User Manual.



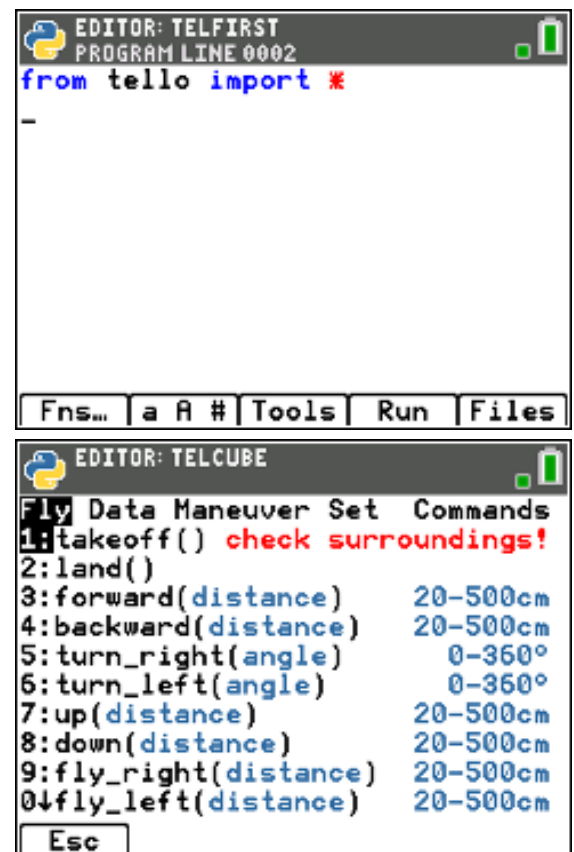
Proceed to the next step to program your TI-84 Plus CE Python graphing calculator to control Tello when your hardware and software are ready.

- From the TI-84 Python File Manager, start a new Python program (ours is called **telfirst.py**) Import the tello module using **<Fns...> Modul <Add-On>** and select  
**from tello import \***

*Note: if you do not see **Tello** on the Add-On Module Import screen then the module is not properly installed in the calculator.*

*Note: you will also see the **tellocfg** module on the **<Add-On>** menu. This module is used to configure the Tello/micro:bit system when necessary.*

- This is the tello module menu system. Note that there are five submenus: **Fly**, **Data**, **Maneuver**, **Set**, and **Commands** that contain all the available Tello methods. You will investigate many (but not all) of these methods in these mini-projects.





### TI-84 PLUS CE PYTHON

3. Make sure the Tello is turned on. There will be a rapidly blinking yellow LED on the front of Tello indicating that Tello is ready to be connected to wi-fi.

Check your connections now by running the program. You should see:

```
Connecting to micro:bit...
Micro:bit is connected
Connecting to Tello 1...
Tello 1 is connected✓
>>>
```

If there is a connection problem, an error message will appear instead. Your Tello may have a different 'name' that was assigned during the setup process. The Tello used here is named '1'.



```
PYTHON SHELL
Connecting to micro:bit...
Micro:bit is connected
Connecting to Tello 1...
Tello 1 is connected
>>> |
```

The screenshot shows a window titled 'PYTHON SHELL' with a Python logo icon. The text inside the window displays the connection process: 'Connecting to micro:bit...', 'Micro:bit is connected', 'Connecting to Tello 1...', 'Tello 1 is connected', and a prompt '>>> |'. The window has a standard OS-style title bar with minimize, maximize, and close buttons. At the bottom, there is a menu bar with 'Fns...', 'a A #', 'Tools', 'Editor', and 'Files'.

4. A good practice is to check the Tello battery charge at the beginning of your programs.

Get the **battery()** function from

**[math] > tello drone... > Data > var=battery()**

And insert a variable name in front of the equals sign: **b** is a common choice, for 'battery'.

Run the program again.

5. The **battery()** function reports the battery charge level as a percentage. If the charge level is too low (<10%) Tello will not fly.

*Note: there's no need to **print(b)**. The **battery()** function, as well as most other Tello methods, displays information for you.*



```
EDITOR: TELFIRST
PROGRAM LINE 0003
from tello import *
b=tello.battery()

PYTHON SHELL
Connecting to micro:bit...
Micro:bit is connected
Connecting to Tello 1...
Tello 1 is connected
BATTERY = 80%
>>> |
```

The screenshot shows two windows. The top window is titled 'EDITOR: TELFIRST' and contains the Python code: 'from tello import \*' and 'b=tello.battery()'. The bottom window is titled 'PYTHON SHELL' and shows the same connection messages as the first screenshot, but with the additional output 'BATTERY = 80%' before the prompt '>>> |'. Both windows have a menu bar at the bottom with 'Fns...', 'a A #', 'Tools', 'Run', and 'Files'.



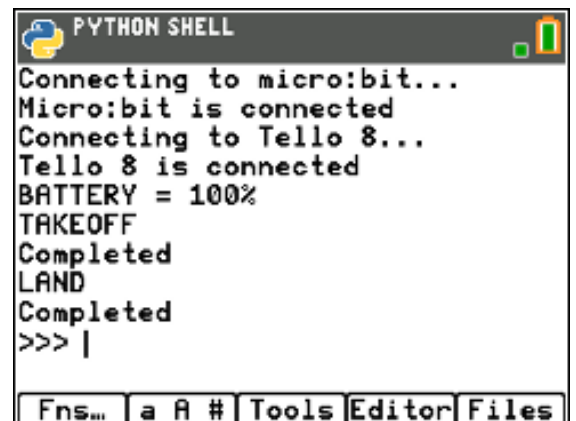
## TI-84 PLUS CE PYTHON

6. If the connections are still successful and there's enough battery charge, then prepare to fly!
- Using the **takeoff()** command causes the Tello to fly upwards about 70 - 80cm (about 27-32 inches). Be sure there is room above the Tello for this maneuver.
- Also, what goes up must come down. We must have full control of the drone at all times, so we must make the Tello **land()** as well.

7. In your program, add these two statements, both found on
- [math] > tello drone... > Tello > Fly**

```
tello.takeoff()  
tello.land()
```

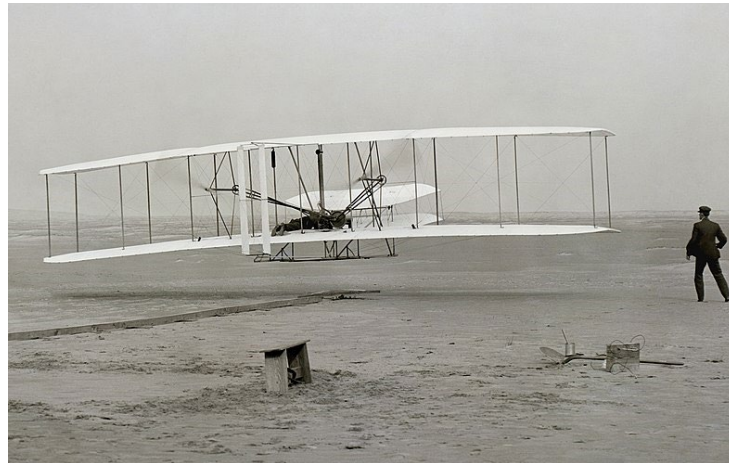
8. Again, make sure all hardware is on, including Tello, and run the program. Your Shell screen will display the previous 'Connecting...' messages and then show messages for each command that is sent to the Tello. The Tello will go up, hover for a few seconds, and then land.
- Tello should land very close to its starting location. If it drifts a little bit that's OK, but if it wanders too far, then Tello is having trouble 'seeing' the ground. Be sure the surface has a colorful pattern and that the area is well-lit.
- Tello communicates with the micro:bit and the TI-84 to confirm that the tasks are properly **Completed**.
- After Tello lands, the program is finished.



## TI-84 PLUS CE PYTHON

- Congratulations on your first successful flight! The Wright Brothers would be proud of you!

Let's try another maneuver...



\* John T. Daniels, Public domain, via Wikimedia Commons

[https://commons.wikimedia.org/wiki/File:First\\_flight2.jpg](https://commons.wikimedia.org/wiki/File:First_flight2.jpg)

- The **[math]** > **tello drone...** > **Fly** menu contains lots of possible maneuvers. Two simple but elegant maneuvers that are relatively *safe* are the **turn\_right()** and **turn\_left()** methods. These keep the Tello in one position and rotate it the given number of degrees.

Try

```
tello.turn_right(360)
tello.turn_left(360)
```

while Tello is in the air. Insert these functions between the **takeoff()** and **land()** methods in your program and enter the number of degrees to turn as each argument.

See the next step for the code...

```
EDITOR: TELFIRST
Fly Data Maneuver Set Commands
1:takeoff() check surroundings!
2:land()
3:forward(distance) 20-500cm
4:backward(distance) 20-500cm
5:turn_right(angle) 0-360°
6:turn_left(angle) 0-360°
7:up(distance) 20-500cm
8:down(distance) 20-500cm
9:fly_right(distance) 20-500cm
0:fly_left(distance) 20-500cm
A:forward_time(time) 1-10s
B:backward_time(time) 1-10s
C:enable_mission_pad(number) ID
D:goto(x,y,z) cm
Esc
```

- Your program should look like the one shown here.

Again, re-check your hardware, including Tello itself and then run the program. Tello turns itself off after a few minutes of inactivity to save the battery.

Tello will take off, spin to the right a full turn, then spin to the left a full turn, then land – hopefully to the same place that it took off!

```
EDITOR: TELFIRST
PROGRAM LINE 0010
from tello import *
b=tello.battery()

tello.takeoff()

tello.turn_right(360)
tello.turn_left(360)

tello.land()
-
Fns... a A # Tools Run Files
```



## TI-84 PLUS CE PYTHON

12. While the program is running, your TI-84 screen will report the progress of each maneuver.

13. Another pair of relatively safe, easy maneuvers are **tello.up( )** and **tello.down( )** also found on the [math] tello drone... > Fly menu.

The arguments, *distance*, are measured in centimeters. Be sure that when flying upward that there is room above to make the maneuver. Tello does not have a camera or sensor on top so it may crash into the ceiling!

14. **Fail:** When using **tello.down( )**, Tello cannot go below about 30cm (about 12 inches) from the surface. Tello does have a sensor on the bottom to measure its height above the 'ground'. In the screen to the right, after takeoff, Tello attempted to go down 100cm as the code requested, but only went down about 50cm, where it reached the 30cm height limit. It hovered there for a while then reported "Fail" and landed the drone. The program's **land( )** function also failed because the drone had already landed.

*There are other situations where an operation can 'Fail'.*

15. Congratulations on your first Tello flying experience! In the subsequent mini-projects, you will experience more of the possible flight maneuvers and practical applications of drones using Tello.

