TI-84 Plus CE Python graphing calculator

Dash Reference Guide v1.0.0

The Dash Reference Guide is a companion document to the Dash Getting Started Guide. This reference documents all of the Python commands within the four Dash modules — ww_dash, dash_out, dash_in, and dash_pth — that must be used to program the Dash Robot. Please see the Getting Started Guide to learn how to set up and write your first Python program using the TI-84 Plus CE Python model.

Wonder Workshop Dash module

Python method	Example code	Notes
Drive menu		
forward(distance)	forward(10)	Drives a distance in the range of 0–80 grid units. 1 unit = 10 cm
backward(distance)	backward(10)	Drives a distance in the range of 0–80 grid units. 1 unit = 10 cm
left(angle)	left(90)	Turns an angle in the range of 0–360 degrees.
right(angle)	right(90)	Turns an angle in the range of 0–360 degrees.
stay(time)	stay(5)	The robot does not move for the duration of time in seconds.
to_xy(x,y)	to_xy(10,10)	Drives shortest distance to x grid units, y grid units position.
to_polar(r,theta)	to_polar(10,45)	Rotates theta degrees counterclockwise from + X-axis and drives r units.
to_angle(angle)	to_angle(270)	Rotates angle degrees counterclockwise from + X-axis.
forward_time(time)	forward_time(6)	It drives for time seconds. Distance traveled is dependent on the speed setting.
backward_time(time)	backward_time(6)	It drives for time seconds. Distance traveled is dependent on the speed setting.
forward(distance,“unit”)	forward(2,“m”)	Drives distance in the specified unit of meters or grid units.
backward(distance,“unit”)	backward(2,“m”)	Drives distance in the specified unit of meters or grid units.
left(angle,“unit”)	left(π/2,“radians”)	Turns an angle of the specified unit, degrees or radians.
right(angle,“unit”)	right(π/2,“radians”)	Turns an angle of the specified unit, degrees or radians.
forward_time(T,S,“unit”)	forward_time(9,.6,“m/s”)	Drives T seconds at speed S with the specified units of m/s or units/s.
backward_time(T,S,“unit”)	backward_time(9,.6,“m/s”)	Drives T seconds at speed S with the specified units of m/s or units/s.
forward(D,“unit”,S,“unit”)	forward(3,“m”,.4,“m/s”)	Drives D distance with the specified unit of m or grid units at speed S in m/s or units/s.
backward(D,“unit”,S,“unit”)	backward(3,“m”,.4,“m/s”)	Drives D distance with the specified unit of m or grid units at speed S in m/s or units/s.

Python method	Example code	Notes
Settings menu		
units/s	forward_time(5,8,“units/s”)	Paste setting into drive methods requiring speed in units/s. Range 1 units/s–10 units/s.
m/s	forward_time(5,.8,“m/s”)	Paste setting into drive methods requiring speed in m/s. Range 0.1 m/s–1.0m/s
units	forward(14,“units”)	Paste setting into drive methods requiring distance in units. Range 0.1 units–80 units.
m	forward(1.4,“m”)	Paste setting into drive methods requiring distance in meters. Range 0.01 m–8.0m.
degrees	left(130,“degrees”)	Paste setting into drive methods requiring angle in degrees. Range 0–360 degrees.
radians	left(π/2,“radians”)	Paste setting into drive methods requiring angle in radians. Range 0–2π.
set_speed(speed)	set_speed(7)	Sets the global speed from 1 to 9 units/sec. The default is 5.
I/O menu		
from dash_in import *	from dash_in import *	Import this module from ww_dash import * to include input methods (for example, button press).
from dash_out import *	from dash_out import *	Import this module from ww_dash import * to include output methods (for example, play a sound).
from dash_pth import *	from dash_pth import *	Import this module from ww_dash import * to include path drive methods (for example, to generate a list of x and y positions of the robot).
Commands menu		
sleep(seconds)	sleep(5)	
disp_at(row,“text”,“align”)	disp_at(10,“Howdy”,“left”)	Displays the specified text in the specified row with the selected justification. Row 1 is the top of the screen, and 11 is the bottom. This method works best when in conjunction with disp_clr(), disp_cursor(0), and disp_wait().
disp_clr()	disp_clr()	It clears the entire display except for the menu bar at the bottom of the screen.
disp_wait()	disp_wait()	Holds the display in the current state until the [clear] key is pressed.
disp_cursor(state)	disp_cursor(state)	Hides the Python REPL (>>>) on the display when state = 0.
while not escape():	while not escape():	A convenient indefinite While statement that loops until the [clear] key is pressed.
position(x,y)	position(40,50)	Translates and sets the origin of Dash’s internal grid. The default origin (0,0) is the location of Dash when it is turned on.
grid_m_unit(scale_value)	grid_m_unit(1)	Redefines the grid unit in terms of meters. In the example, the grid unit is redefined from the default of 1 unit = .1 m to 1 unit = 1m.

Dash Input module

Python method	Example code	Notes
Input menu		
from dash_in import *	from dash_in import *	This statement must be added to the program to enable all input methods.
wait_for_press(“button”)		This method will pause program execution and wait for the selected button to be pressed.
wait_for_clap()		This method will pause program execution and wait for a loud hand clap to be heard by Dash.
is_pressed(“button”)		This method returns True if the selected button is pushed at the moment when the method is executed. If the button is not pressed, the method will return False.
was_pressed(“button”)		This method returns True if the specified button has been pushed at any time since the method was last used. Using this method will revert to return to False.
is_clap()		This method returns True if Dash hears a clap at the moment when the method is executed. If no clap is heard, the method will return False.
is_close(“direction”)		This method uses the front-facing IR sensor. True is returned if the Dash is close to an object in the specified direction, and False when no object is detected.
Buttons menu		“top”, “one”, “two”, “three”
Directions		“in_front”, “behind”, “left”, “right”

Dash Output module

Python method	Example code	Notes
Output menu		
from dash_out import *	from dash_out import *	This statement must be added to the program to enable all output methods.
set_light(“light”,“color”)	set_light(“front”,“orange”)	Turns on Dash’s front-facing LED orange. Turn off an LED, and use “off”.
set_eye_pattern(“pattern”)	set_eye_pattern(“happy”)	Sets the 16 Dash eye LEDs in the pattern selected from the Eye menu.
set_eye_positions(position)	set_eye_positions(0b1010101010101010)	Sets the 16 Dash eye LEDs in the pattern specified in a 16-bit binary number. The digit 1 is on, and 0 is off. In the example code, every other LED is turned on.
eye_brightness(value)	eye_brightness(128)	Sets all of the Dash eye LEDs to the specified brightness. 0 is off, and 255 is the brightest.
play_sound(“sound”)	play_sound(“Dinosaur”)	Plays the specified sound. The play sound will not block the drive function.
play_tone(frequency,duration)	play_tone(440,5)	It plays the specified tone in Hz, which must be greater than 50Hz. The example plays tuning A for five seconds.
set_volume(loudness)	set_volume(4)	Sets the volume of the Dash speaker. 0 will turn off the speaker, and 6 is the loudest.
look_forward()	look_forward()	Turns Dash’s head forward and tilts head to level.
look_left(angle)	look_left(45)	Turns Dash’s head left at the specified angle. Range of motion is 0–130 degrees.
look_right(angle)	look_right(90)	Turns Dash’s head right at the specified angle. Range of motion is 0–130 degrees.
look_up(angle)	look_up(15)	Tilts Dash’s head up the specified angle. Range of motion is 0–22 degrees.
look_down(angle)	look_down(20)	Tilts Dash’s head down the specified angle. Range of motion is 0–7 degrees.
Lights menu		“all”, “left”, “right”, “front”
Colors menu		“red”, “green”, “blue”, “yellow”, “cyan”, “magenta”, “orange”, “white”, “off”
Sounds menu		“Horse”, “Cat”, “Dog”, “Dinosaur”, “Lion”, “Goat”, “Crocodile”, “Elephant”, “Beeps”, “Lasers”, “Gobble”, “Buzz”, “Ay Yai”, “Squeak”, “Hi”, “Huh”, “Uh Oh”, “Okay”, “Sigh”, “Tada”, “Wee”, “Bye”, “Fire Siren”
Eye menu		“on”, “off”, “happy”, “upside_down_happy”, “brow”, “alternate_left”, “alternate_right”

Dash Path module

Python method	Example code	Notes
Path menu		
from dash_pth import *	from dash_pth import *	This statement must be added to the program to enable all path methods.
pathlist_x()	x = path_x()	This method logs and returns a list with the x-grid position coordinate of each x,y waypoint when Dash drives a path with more than one drive command.
pathlist_y()	y = path_y()	This method logs and returns a list with the y-grid position coordinate of each x,y waypoint when Dash drives a path with more than one drive command.
pathlist_time()	t = path_time()	This method logs and returns a list with the time stamps of each x,y waypoint when Dash drives a path with more than one drive command.
pathlist_heading()	h = path_heading()	This method logs and returns a list of the angular headings of each x,y waypoint when Dash drives a path with more than one drive command. Note that when Dash is turned on, a virtual grid is created with the direction Dash is pointing set as the zero-degree heading along the virtual positive x-axis. Angle is measured counterclockwise from the positive x-axis.
pathlist_distance()	d = path_distance()	This method logs and returns a list of the distances, in grid units, among each x,y waypoint when Dash drives a path with more than one drive command.
waypoint_x()	x = waypoint_x()	This method returns a floating point value of Dash's x-coordinate of the most recent waypoint.
waypoint_y()	y = waypoint_y()	This method returns a floating point value of Dash's y-coordinate of the most recent waypoint.
waypoint_heading()	h = waypoint_heading()	This method returns a floating point value of Dash's angular heading of the most recent waypoint.
Commands menu		
store_list("name",var)	store_list("1",x) store_list("2",y)	This command stores a Python list to the TI-84 CE Plus operating system's reserved list variables L1–L6. The "name" of the list is the number of the list variable (for example, variable L1 is entered as "1"). Acceptable range of value is 1–6. Once the Python list is exported to the operating system as L1–L6, the lists may be used in the calculator's stat plot application to graph the path Dash drove in the Python program.
path_clear()	path_clear()	Clears the Dash's lists from memory. This will automatically occur when a program is rerun.