GoPiGo Python Tutorials

Contents

NOTE: All Python code is compatible with Python 3.5. This is the current version of Python on the GoPiGo.

Go to https://gopigo3.readthedocs.io/en/master/api-basic/easygopigo3.html#easygopigo3 for information on the easygopigo3 library.

First Steps

Go to the Code Examples folder in the WNCCNASA GitHub repository. Copy and paste the example code to the GoPiGo to get started.

Easy Movement

Now it is time to do some tutorials to learn the GoPiGo library better.

Learning points

- Functions, Loops, Movement, GoPiGo Blinkers

```python
#!/usr/bin/env python3
"""
    Name: easy_movement.py
    Author: William A Loring
    Created: 09-18-21 Revised:
    Purpose: Demonstrate a sampling of GoPiGo dead reckoning movements
"""
# This uses the EasyGoPiGo3 library.  You can find more information on the li
# here:  https://gopigo3.readthedocs.io/en/master/api-basic/easygopigo3.html#

# Import the time library for the sleep function
import time
# Import GoPiGo3 library
from easygopigo3 import EasyGoPiGo3

# Create an instance of the GoPiGo3 class
# GPG is the GoPiGo3 object used to access methods and properties
gpg = EasyGoPiGo3()


#------------------------- SQUARE RIGHT ------------------------#
def square_right(distance):
    """
        Drive a right square based on the distance argument
    """
    # Loop four times,  Loop starts at 0,
    # Ends at 1 less than the last number
    # The loop increments 0, 1, 2, 3
    print("Square Right")
    for x in range(0, 4):
        # Print the loop counter
        print(x)
        gpg.led_off("right")
        gpg.drive_inches(
            distance,   # How far to drive in inches
            True        # Blocking, nothing else can happen while moving
        )
        gpg.led_on("right")
        # Turn right 90 degrees, positive number is right
        gpg.turn_degrees(90)
    # Turn both blinkers off
    gpg.led_off("right")
    gpg.led_off("left")


#------------------------- SQUARE LEFT ------------------------#
def square_left(distance):
    """
        Drive a left square based on the distance argument
    """
    print("Square Left")
    for x in range(0, 4):
        print(x)
        gpg.led_off("left")
        gpg.drive_inches(distance, True)
        gpg.led_on("left")
        # Turn left 90 degrees, - is left
        gpg.turn_degrees(-90)
    gpg.led_off("left")

```

```
#------------------------- GOPIGO WAGGLE ----------------------#
def waggle():
    """ Waggle back and forth """
    print("Waggle")
    for x in range(0, 4):
        print(x)
        gpg.led_on("left")
        gpg.turn_degrees(-10)
        gpg.led_off("left")
        gpg.led_on("right")
        gpg.turn_degrees(10)
        gpg.led_off("right")
    # Turn off both blinkers
    gpg.led_off("right")
    gpg.led_off("right")


def main():
    """ Main Program Entry Point """
    # Drive a 5" square turning left
    square_left(5)

    # Turn left to reverse the square
    print("Turn Left 90")
    gpg.turn_degrees(-90)

    # Drive a 5" square turning right
    square_right(5)

    print("Spin left.")
    gpg.spin_left()
    time.sleep(1)

    # Waggle back and forth
    waggle()

    print("Spin right.")
    gpg.spin_right()
    time.sleep(3)

    print("Stop!")
    gpg.stop()
    print("Done!")


# If a standalone program, call the main function
# Else, use as a module
if __name__ == '__main__':
    main()
```