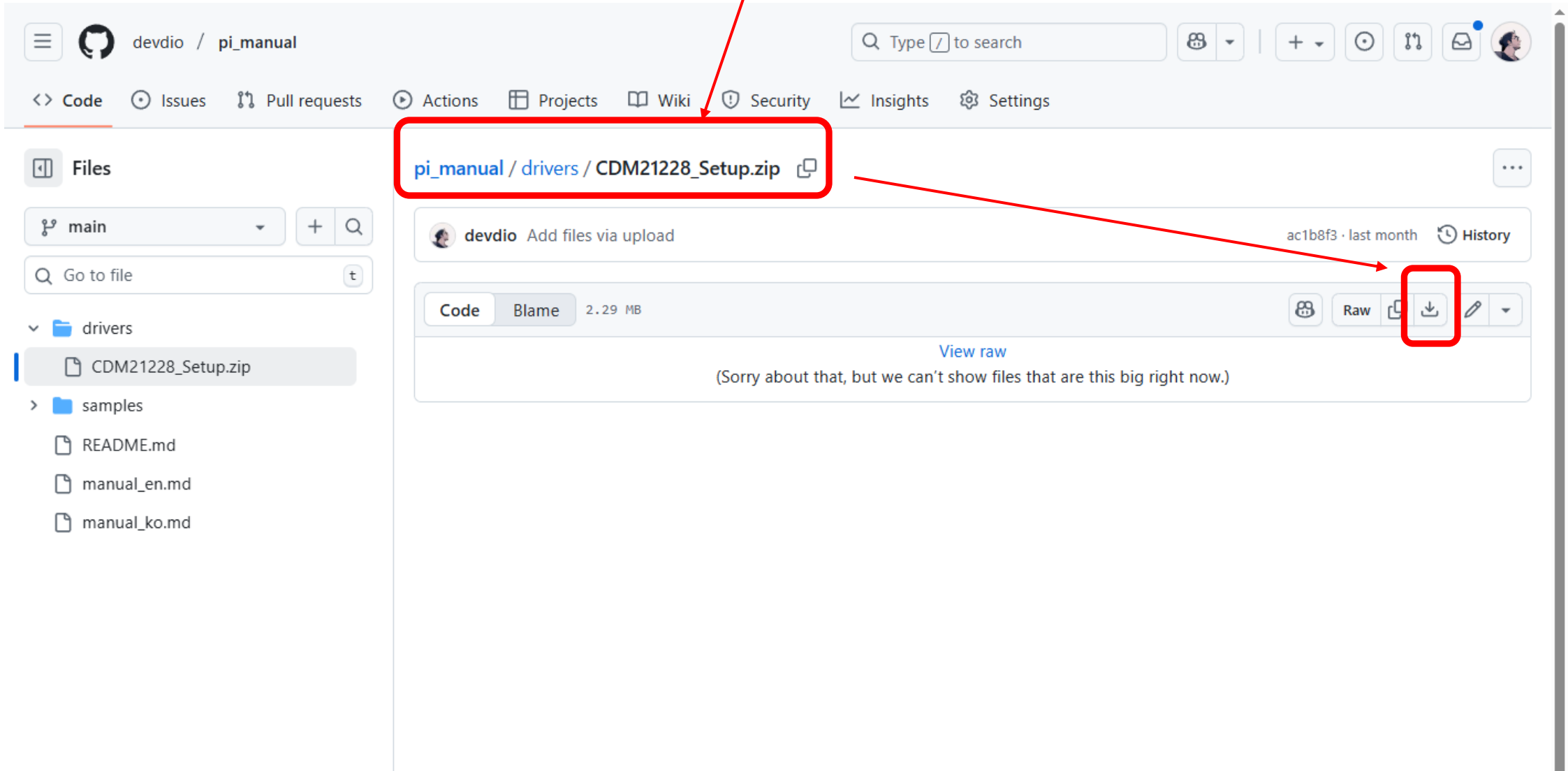
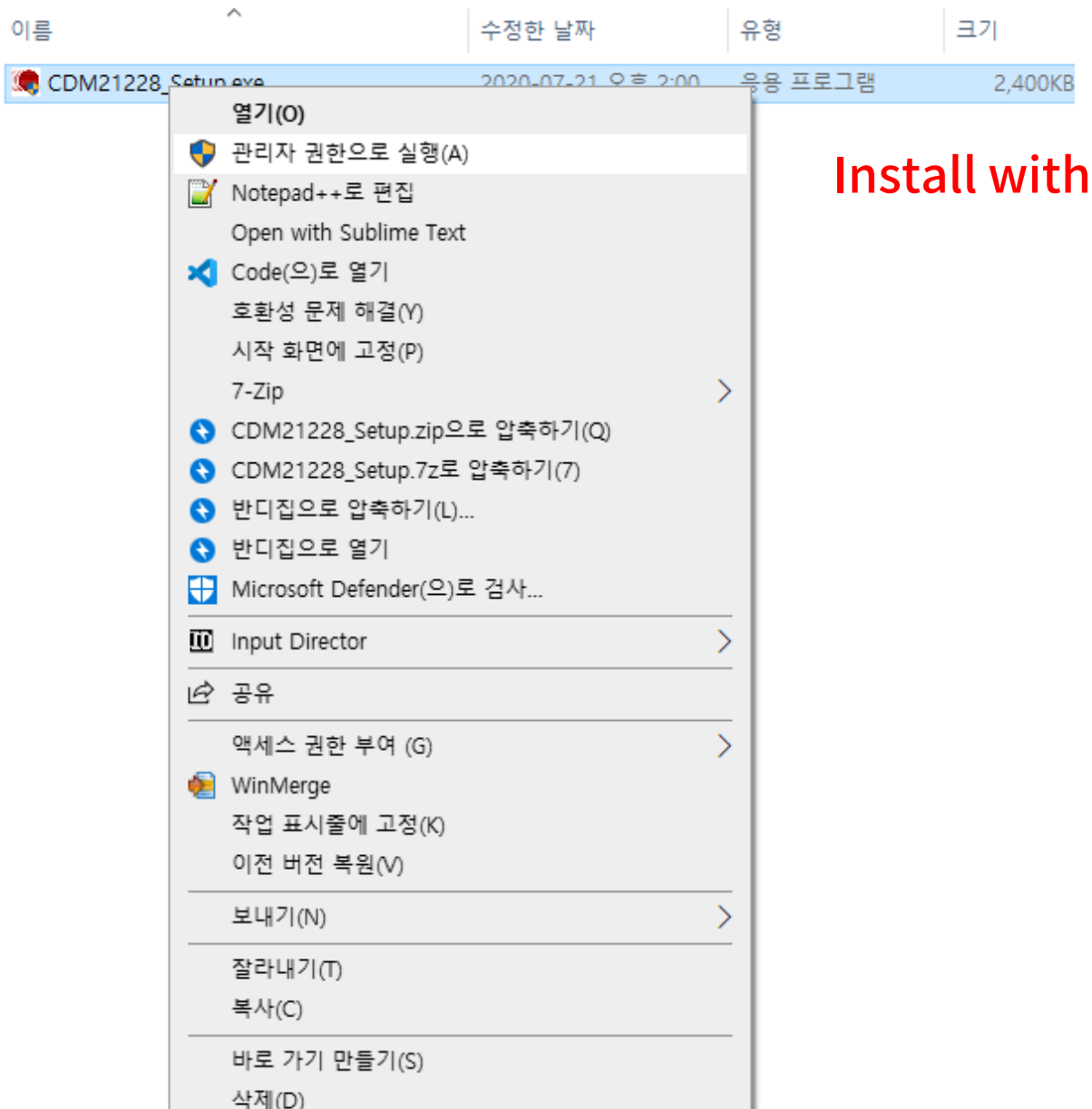
KamibotPI

Install USB Driver

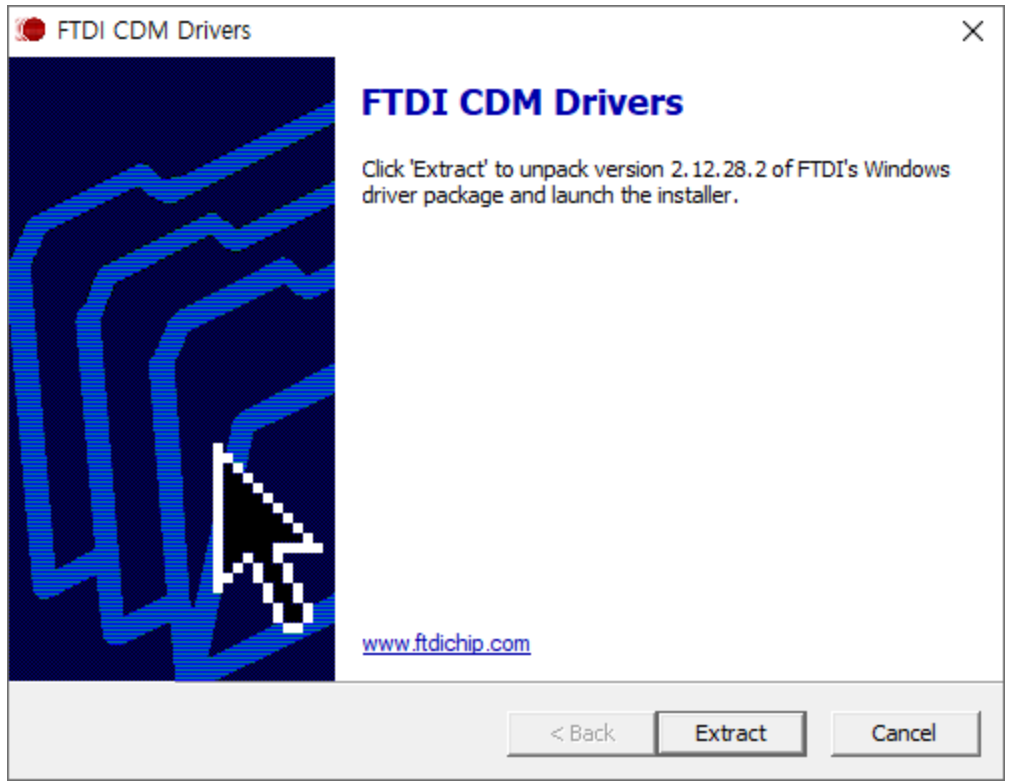
https://github.com/devdio/pi_manual/



Install USB Driver

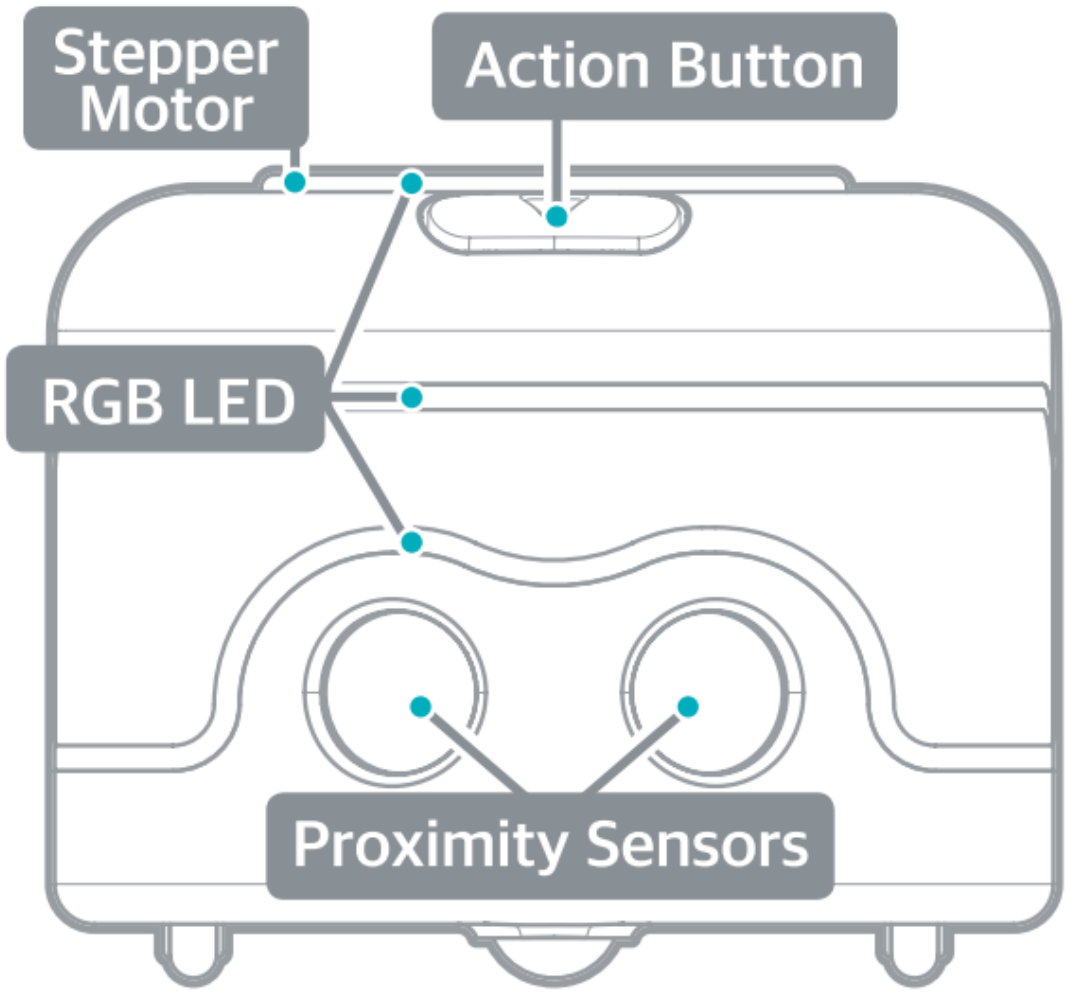


Install with Windows administrator privileges

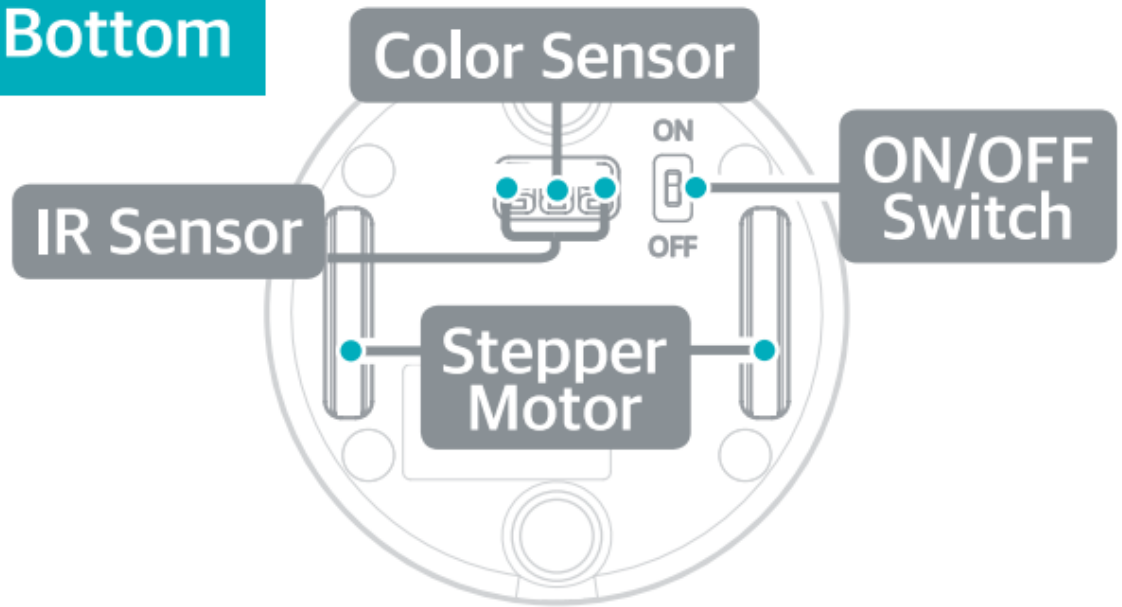

FTDI CDM Drivers
FTDI CDM Drivers
Click 'Extract' to unpack version 2.12.28.2 of FTDI's Windows driver package and launch the installer.
www.ftdichip.com
< Back
Extract
Cancel


장치 드라이버 설치 마법사
장치 드라이버 설치 마법사 완료
컴퓨터에 드라이버를 설치했습니다.
이제 장치를 컴퓨터에 연결할 수 있습니다. 장치와 함께 설명서가 제공되었으면 장치를 사용하기 전에 설명서를 먼저 읽어보십시오.
드라이버 이름
상태
FTDI CDM Driver Pa...
사용할 수 있음
FTDI CDM Driver Pa...
사용할 수 있음
< 뒤로(B)
마침
취소

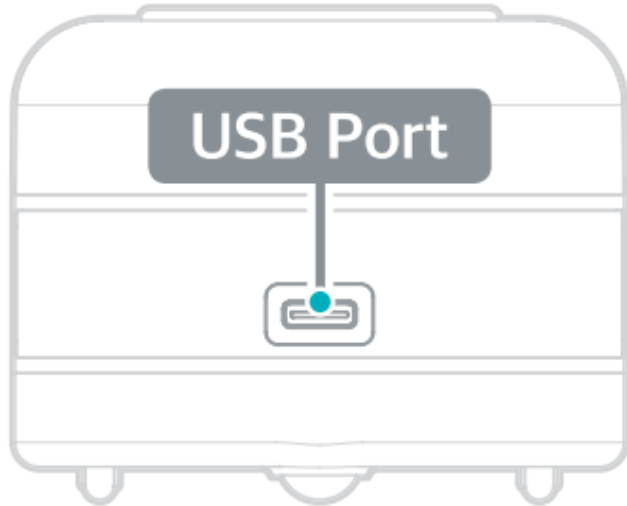
Front



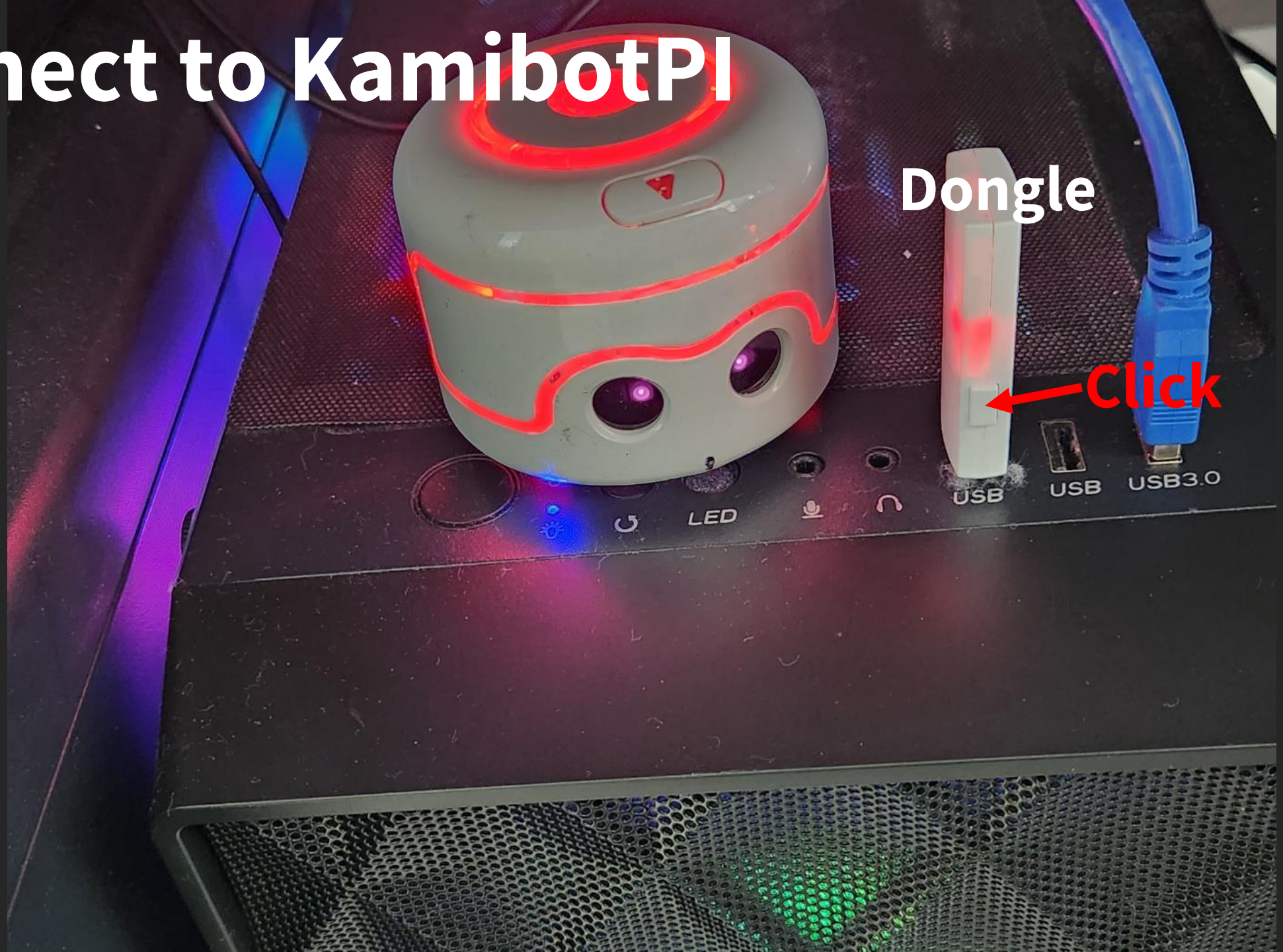
Bottom



Rear

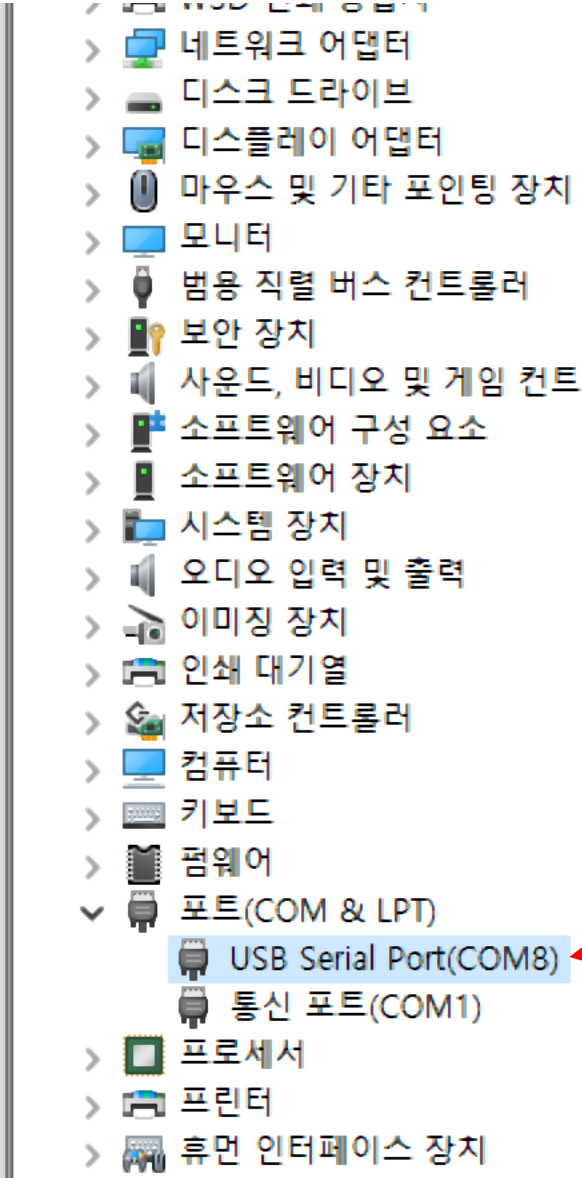


Connect to KamibotPI



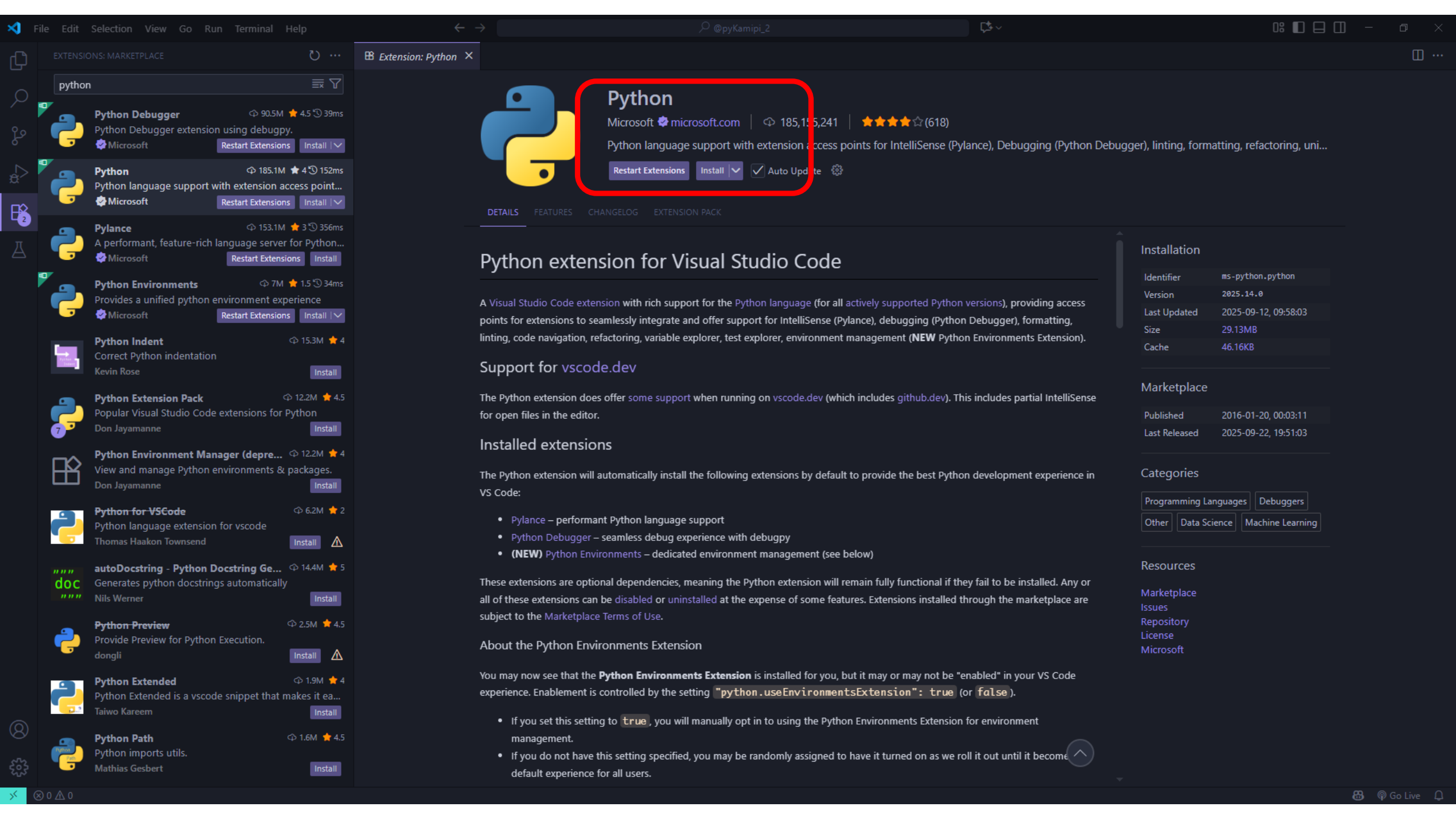
The LED turns Blue upon connection.

Check COM Port

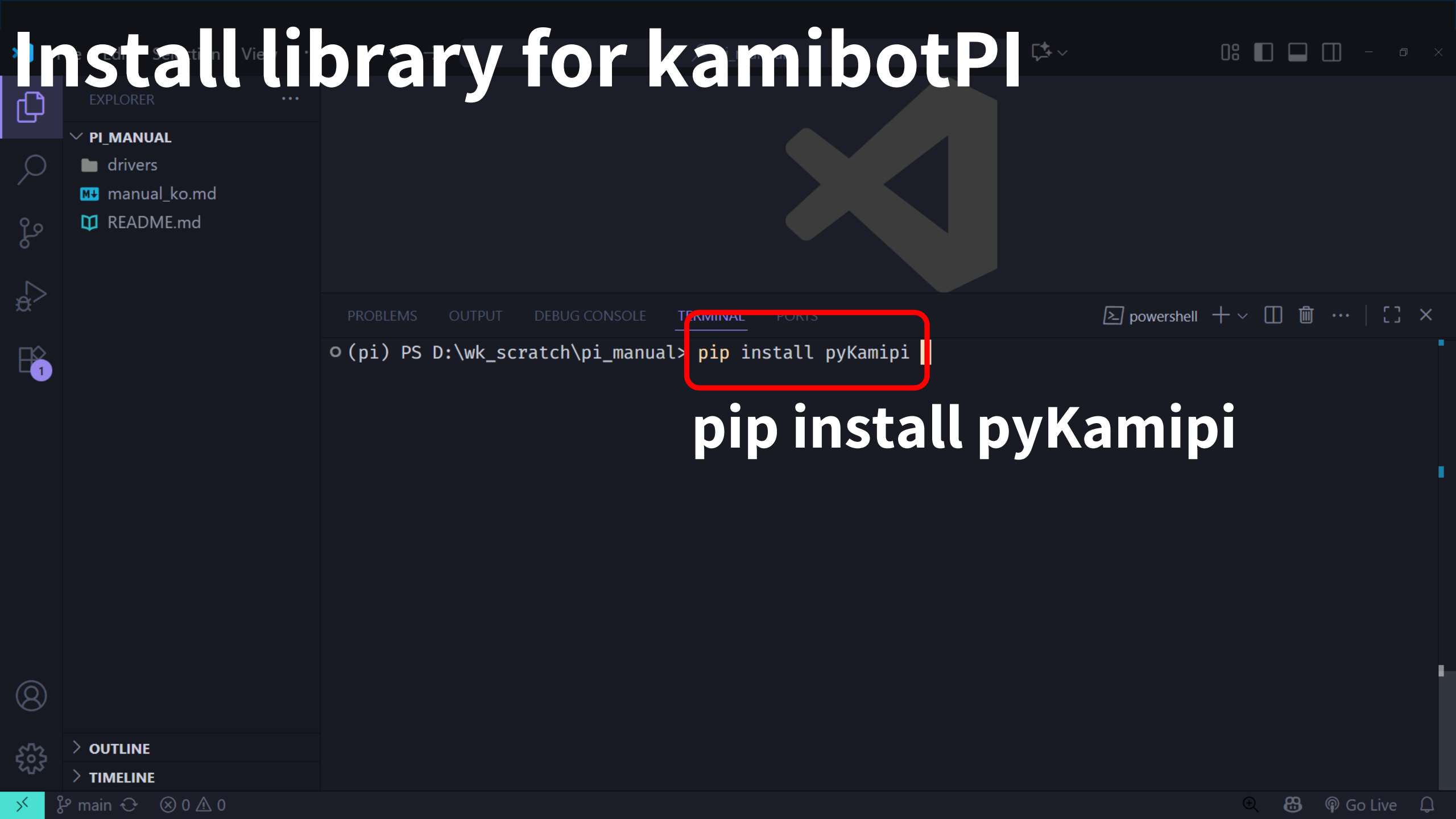


COM Port

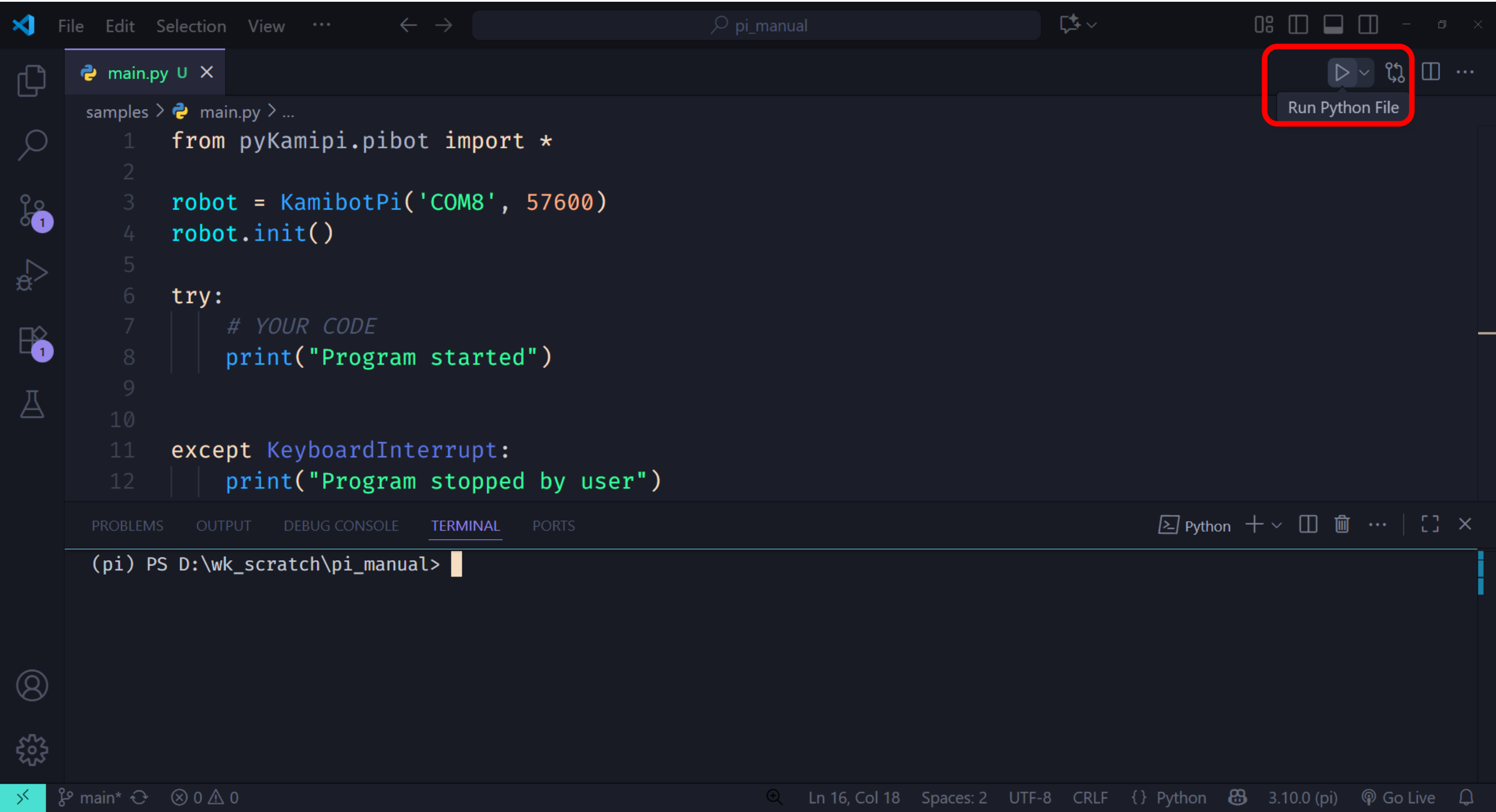
The COM port number may vary depending on your PC
Remember the COM port number

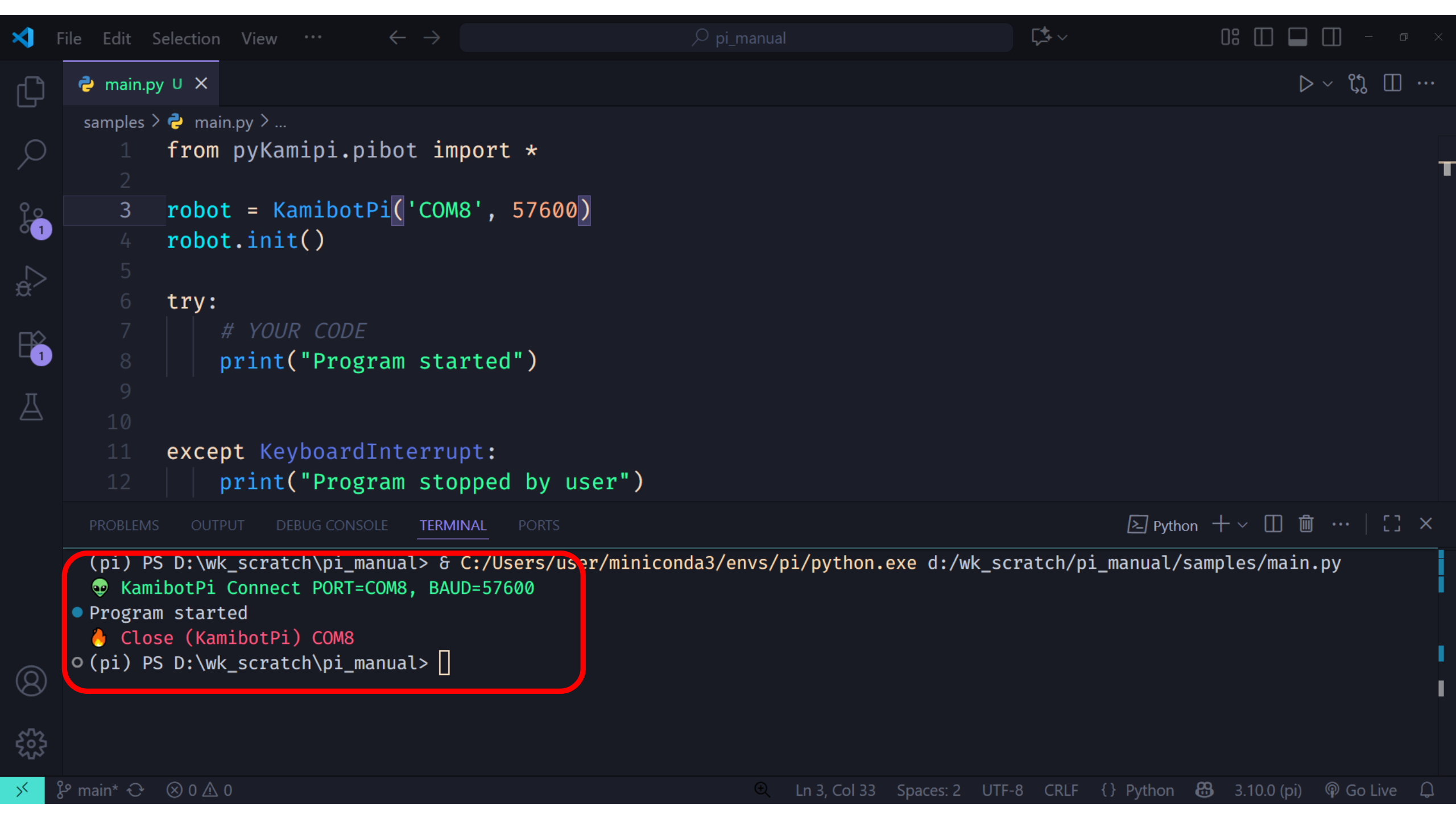

File Edit Selection View Go Run Terminal Help
@pyKamipi_2
EXTENSIONS: MARKETPLACE
python
Python Debugger 90.5M 4.5 39ms
Python Debugger extension using debugpy.
Microsoft Restart Extensions Install
Python 185.1M 4 152ms
Python language support with extension access point...
Microsoft Restart Extensions Install
Pylance 153.1M 3 356ms
A performant, feature-rich language server for Python...
Microsoft Restart Extensions Install
Python Environments 7M 1.5 34ms
Provides a unified python environment experience
Microsoft Restart Extensions Install
Python Indent 15.3M 4
Correct Python indentation
Kevin Rose Install
Python Extension Pack 12.2M 4.5
Popular Visual Studio Code extensions for Python
Don Jayamanne Install
Python Environment Manager (depre... 12.2M 4
View and manage Python environments & packages.
Don Jayamanne Install
Python for VSCode 6.2M 2
Python language extension for vscode
Thomas Haakon Townsend Install
autoDocstring - Python Docstring Ge... 14.4M 5
Generates python docstrings automatically
Nils Werner Install
Python Preview 2.5M 4.5
Provide Preview for Python Execution.
dongli Install
Python Extended 1.9M 4
Python Extended is a vscode snippet that makes it ea...
Taiwo Kareem Install
Python Path 1.6M 4.5
Python imports utils.
Mathias Gesbert Install
Extension: Python
Python
Microsoft microsoft.com 185,155,241 (618)
Python language support with extension access points for IntelliSense (Pylance), Debugging (Python Debugger), linting, formatting, refactoring, uni...
Restart Extensions Install Auto Update
DETAILS FEATURES CHANGELOG EXTENSION PACK
Python extension for Visual Studio Code
A Visual Studio Code extension with rich support for the Python language (for all actively supported Python versions), providing access points for extensions to seamlessly integrate and offer support for IntelliSense (Pylance), debugging (Python Debugger), formatting, linting, code navigation, refactoring, variable explorer, test explorer, environment management (NEW Python Environments Extension).
Support for vscode.dev
The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.
Installed extensions
The Python extension will automatically install the following extensions by default to provide the best Python development experience in VS Code:
Pylance – performant Python language support
Python Debugger – seamless debug experience with debugpy
(NEW) Python Environments – dedicated environment management (see below)
These extensions are optional dependencies, meaning the Python extension will remain fully functional if they fail to be installed. Any or all of these extensions can be disabled or uninstalled at the expense of some features. Extensions installed through the marketplace are subject to the Marketplace Terms of Use.
About the Python Environments Extension
You may now see that the Python Environments Extension is installed for you, but it may or may not be "enabled" in your VS Code experience. Enablement is controlled by the setting "python.useEnvironmentsExtension": true (or false).
If you set this setting to true, you will manually opt in to using the Python Environments Extension for environment management.
If you do not have this setting specified, you may be randomly assigned to have it turned on as we roll it out until it become
default experience for all users.
Installation
Identifier ms-python.python
Version 2025.14.0
Last Updated 2025-09-12, 09:58:03
Size 29.13MB
Cache 46.16KB
Marketplace
Published 2016-01-20, 00:03:11
Last Released 2025-09-22, 19:51:03
Categories
Programming Languages Debuggers
Other Data Science Machine Learning
Resources
Marketplace
Issues
Repository
License
Microsoft
Go Live

Install library for kamibotPI



pip install pyKamipi


File
Edit
Selection
View
pi_manual
main.py U
Run Python File
samples > main.py > ...
1 from pyKamipi.pibot import *
2
3 robot = KamibotPi('COM8', 57600)
4 robot.init()
5
6 try:
7 # YOUR CODE
8 print("Program started")
9
10
11 except KeyboardInterrupt:
12 print("Program stopped by user")
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
(pi) PS D:\wk_scratch\pi_manual>
main*
0 0
Ln 16, Col 18
Spaces: 2
UTF-8
CRLF
Python
3.10.0 (pi)
Go Live


File Edit Selection View ···
pi_manual
main.py U
samples > main.py > ...
1 from pyKamipi.pibot import *
2
3 robot = KamibotPi('COM8', 57600)
4 robot.init()
5
6 try:
7 # YOUR CODE
8 print("Program started")
9
10
11 except KeyboardInterrupt:
12 print("Program stopped by user")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
(pi) PS D:\wk_scratch\pi_manual> & C:/Users/user/miniconda3/envs/pi/python.exe d:/wk_scratch/pi_manual/samples/main.py
KamibotPi Connect PORT=COM8, BAUD=57600
Program started
Close (KamibotPi) COM8
(pi) PS D:\wk_scratch\pi_manual>
main* 0 0
Ln 3, Col 33 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live

Control the direction and speed


Left
Right

Right Wheel

Left Wheel

`go_dir_speed(ldir, lspeed, rdir, rspeed)`

Controls left and right wheel direction and speed individually.

Parameters:

- `ldir` (str): Left wheel rotation direction
  - `'f'`: Forward
  - `'b'`: Backward
- `lspeed` (int): Left wheel speed (0-255)
- `rdir` (str): Right wheel rotation direction
  - `'f'`: Forward
  - `'b'`: Backward
- `rspeed` (int): Right wheel speed (0-255)

Returns: None

Example:

```python
robot.go_dir_speed('f', 100, 'f', 100)  # Both wheels forward at speed 100
robot.go_dir_speed('f', 150, 'b', 150)  # Left forward, right backward
```

File Edit Selection View ··· pi_manual

main.py U

samples > main.py > ...

```python
from pyKamipi.pibot import *

robot = KamibotPi('COM8', 57600)
robot.init()

try:
    # YOUR CODE
    robot.go_dir_speed('f', 100, 'f', 100)
    robot.delayms(2000)
    robot.stop()

except KeyboardInterrupt:
    print("Program stopped by user")

finally:
    robot.stop()
    robot.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python

```
es/main.py
(pi) PS D:\wk_scratch\pi_manual>
```

main* 0 0 Ln 13, Col 37 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live

Precision Control

`move_step(ldir, lstep, rdir, rstep)`

Moves precisely in step units.

Parameters:

- `ldir` (str): Left wheel rotation direction ('f': forward, 'b': backward)
- `lstep` (int): Left wheel step count
- `rdir` (str): Right wheel rotation direction ('f': forward, 'b': backward)
- `rstep` (int): Right wheel step count

Returns: None

Example:

```python
robot.move_step('f', 100, 'f', 100)  # Move forward 100 steps
robot.move_step('f', 200, 'b', 200)  # Rotate in place
```

Let's each calculate the distance of 1 Step


File Edit Selection View ···
pi_manual
main.py U
samples > main.py > ...
2
robot = KamibotPi('COM8', 57600)
robot.init()
try:
# YOUR CODE
robot.move_step('f', 100, 'f', 100)
except KeyboardInterrupt:
print("Program stopped by user")
finally:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
(pi) PS D:\wk_scratch\pi_manual>
main* 0 0
Ln 2, Col 1 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live


File Edit Selection View ···
pi_manual
main.py U
samples > main.py > ...
robot = KamibotPi('COM8', 57600)
robot.init()
try:
# YOUR CODE
robot.move_step('f', 90, 'b', 90)
except KeyboardInterrupt:
print("Program stopped by user")
finally:
robot.stop()
robot.close()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
(pi) PS D:\wk_scratch\pi_manual>
main* 0 0 Ln 8, Col 37 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live

`move_time(ldir, lsec, rdir, rsec)`

Moves in time units.

Parameters:

- `ldir` (str): Left wheel rotation direction ('f': forward, 'b': backward)
- `lsec` (int): Left wheel operation time (seconds)
- `rdir` (str): Right wheel rotation direction ('f': forward, 'b': backward)
- `rsec` (int): Right wheel operation time (seconds)

Returns: None

Example:

```python
robot.move_time('f', 2, 'f', 2)  # Move forward for 2 seconds
robot.move_time('f', 1, 'b', 1)  # Rotate in place for 1 second
```

File Edit Selection View ··· pi_manual

main.py U

samples > main.py > ...

```python
robot = KamibotPi('COM8', 57600)
robot.init()

try:
    # YOUR CODE
    robot.move_time('f', 2, 'f', 2)

except KeyboardInterrupt:
    print("Program stopped by user")

finally:
    robot.stop()
    robot.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python

```
(pi) PS D:\wk_scratch\pi_manual>
```

main* 0 0 Ln 14, Col 17 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live

`move_forward_unit(value, opt, speed)`

Moves forward with specified units.

Parameters:

- `value` (int): Value to move
- `opt` (str, optional): Unit type
  - `'-l'`: cm (default)
  - `'-t'`: seconds
  - `'-s'`: steps
- `speed` (int, optional): Movement speed (0-255), default 50

Returns: None

Example:

```python
robot.move_forward_unit(10, '-l', 50)  # Move forward 10cm
robot.move_forward_unit(2, '-t', 100)  # Move forward for 2 seconds
robot.move_forward_unit(500, '-s', 80)  # Move forward 500 steps
```

File Edit Selection View ··· pi_manual

main.py U

samples > main.py > ...

```python
robot = KamibotPi('COM8', 57600)
robot.init()

try:
    # YOUR CODE
    robot.move_forward_unit(400, '-s', 50)
    # robot.move_forward_unit(400, '-s', 150)

except KeyboardInterrupt:
    print("Program stopped by user")

finally:
    robot.stop()
    robot.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python

```
(pi) PS D:\wk_scratch\pi_manual>
```

main* 0 0 Ln 7, Col 16 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live

`move_backward_unit(value, opt, speed)`

Moves backward with specified units.

Parameters:

- `value` (int): Value to move
- `opt` (str, optional): Unit type ('-l': cm, '-t': seconds, '-s': steps), default '-l'
- `speed` (int, optional): Movement speed (0-255), default 50

Returns: None

Example:

```python
robot.move_backward_unit(10, '-l', 50)  # Move backward 10cm
```

`turn_left_speed(value, speed)`

Rotates left in place by specified angle.

Parameters:

- `value` (int, optional): Rotation angle, default 90
- `speed` (int, optional): Rotation speed (0-255), default 50

Returns: None

Example:

```python
robot.turn_left_speed(90, 50)  # Turn left 90 degrees
robot.turn_left_speed(180, 80)  # Turn left 180 degrees
```

`turn_right_speed(value, speed)`

Rotates right in place by specified angle.

Parameters:

- `value` (int, optional): Rotation angle, default 90
- `speed` (int, optional): Rotation speed (0-255), default 50

Returns: None

Example:

```python
robot.turn_right_speed(90, 50)  # Turn right 90 degrees
robot.turn_right_speed(45, 30)  # Turn right 45 degrees
```

LED Control

`turn_led(rval, gval, bval)`

Sets LED color with RGB values.

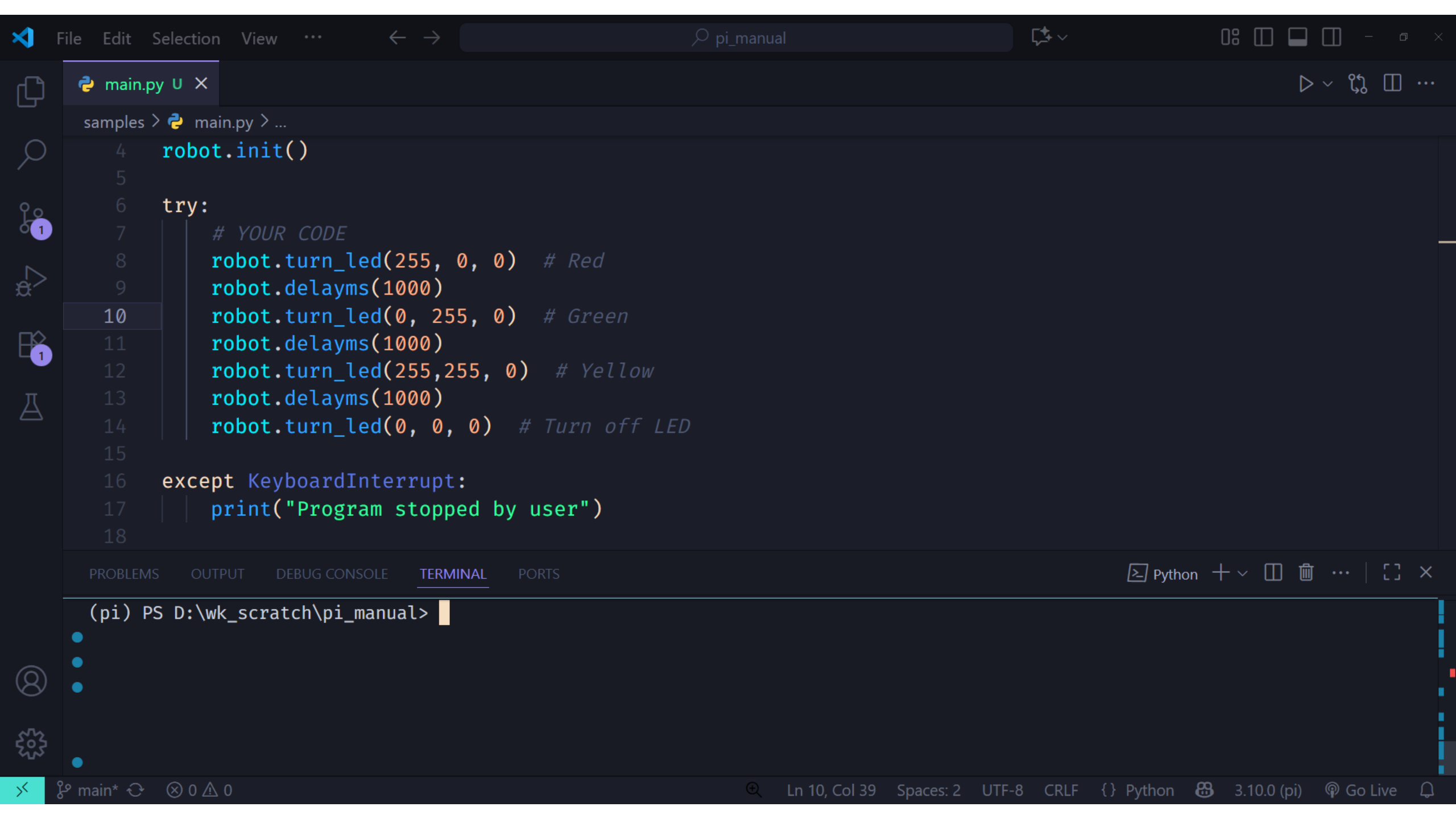
Parameters:

- `rval` (int): Red value (0-255)
- `gval` (int): Green value (0-255)
- `bval` (int): Blue value (0-255)

Returns: None

Example:

```python
robot.turn_led(255, 0, 0)  # Red
robot.turn_led(0, 255, 0)  # Green
robot.turn_led(255, 255, 0)  # Yellow
robot.turn_led(0, 0, 0)  # Turn off LED
```


File Edit Selection View ···
pi_manual
main.py U
samples > main.py > ...
4 robot.init()
5
6 try:
7 # YOUR CODE
8 robot.turn_led(255, 0, 0) # Red
9 robot.delayms(1000)
10 robot.turn_led(0, 255, 0) # Green
11 robot.delayms(1000)
12 robot.turn_led(255,255, 0) # Yellow
13 robot.delayms(1000)
14 robot.turn_led(0, 0, 0) # Turn off LED
15
16 except KeyboardInterrupt:
17 print("Program stopped by user")
18
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
(pi) PS D:\wk_scratch\pi_manual>
main* 0 0
Ln 10, Col 39 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live

Top Motor Control

`top_motor_degree(dir, value, speed)`

Rotates the top motor by specified angle.

Parameters:

- `dir` (str): Rotation direction
  - `'l'`: Left
  - `'r'`: Right
- `value` (int, optional): Rotation angle, default 90
- `speed` (int, optional): Rotation speed (0-255), default 50

Returns: None

Example:

```python
robot.top_motor_degree('r', 90, 50)  # Rotate right 90 degrees
robot.top_motor_degree('l', 180, 80)  # Rotate left 180 degrees
```

File Edit Selection View ··· pi_manual

main.py U

samples > main.py > ...

```python
robot = KamibotPi('COM8', 57600)
robot.init()

try:
    # YOUR CODE
    robot.top_motor_degree('r', 90, 50)
    robot.top_motor_degree('l', 180, 80)

except KeyboardInterrupt:
    print("Program stopped by user")

finally:
    robot.stop()
    robot.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python

```
(pi) PS D:\wk_scratch\pi_manual>
```

main* 0 0 Ln 9, Col 5 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live

`top_motor_abspos(degree, speed)`

Moves the top motor to an absolute angle position.

Parameters:

- `degree` (int, optional): Absolute angle position (0-65000), default 0
- `speed` (int, optional): Rotation speed (0-255), default 50

Returns: None

Example:

```python
robot.top_motor_abspos(0, 50)  # Move to 0 degree position
robot.top_motor_abspos(180, 80)  # Move to 180 degree position
```

`top_motor_time(dir, value, speed)`

Rotates the top motor for specified time.

Parameters:

- `dir` (str): Rotation direction ('l': left, 'r': right)
- `value` (int, optional): Rotation time (seconds), default 3
- `speed` (int, optional): Rotation speed (0-255), default 50

Returns: None

Example:

```python
robot.top_motor_time('r', 2, 50)  # Rotate right for 2 seconds
robot.top_motor_time('l', 5, 80)  # Rotate left for 5 seconds
```

`top_motor_round(dir, value, speed)`

Rotates the top motor by specified number of rotations.

Parameters:

- `dir` (str): Rotation direction ('l': left, 'r': right)
- `value` (int, optional): Number of rotations, default 1
- `speed` (int, optional): Rotation speed (0-255), default 50

Returns: None

Example:

```python
robot.top_motor_round('r', 2, 50)  # Rotate right 2 times
robot.top_motor_round('l', 1, 100)  # Rotate left 1 time
```

`top_motor_stop()`

Immediately stops the top motor rotation.

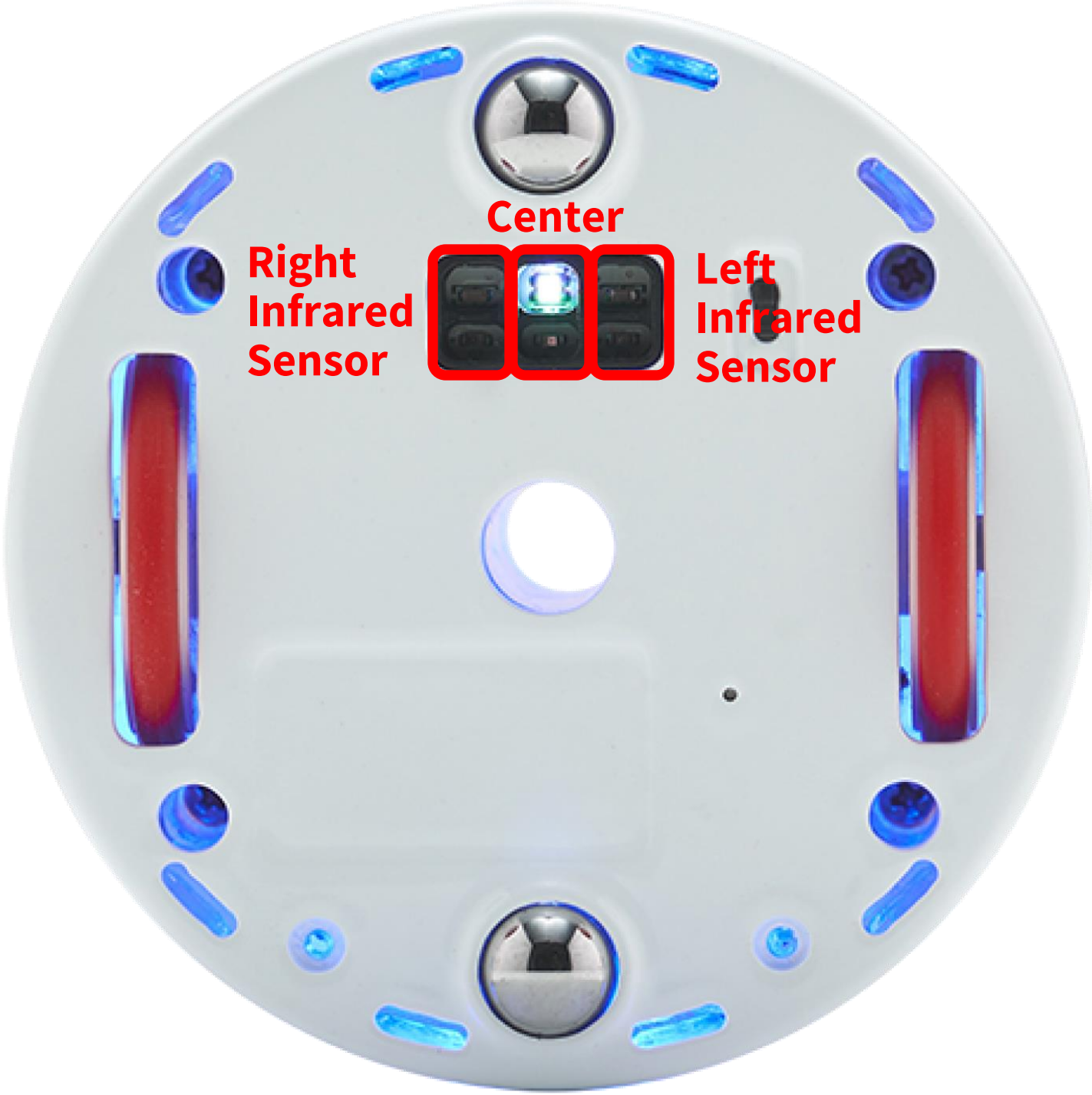
Parameters: None

Returns: None

Example:

```python
robot.top_motor_stop()
```

Line Sensor


Center
Right
Infrared
Sensor
Left
Infrared
Sensor

`get_line_sensor(opt)`

Activates the line sensor and reads values.

Parameters:

- `opt` (bool, optional): Sensor operation mode
  - `True`: Activate sensor (default)
  - `False`: Stop sensor

Returns:

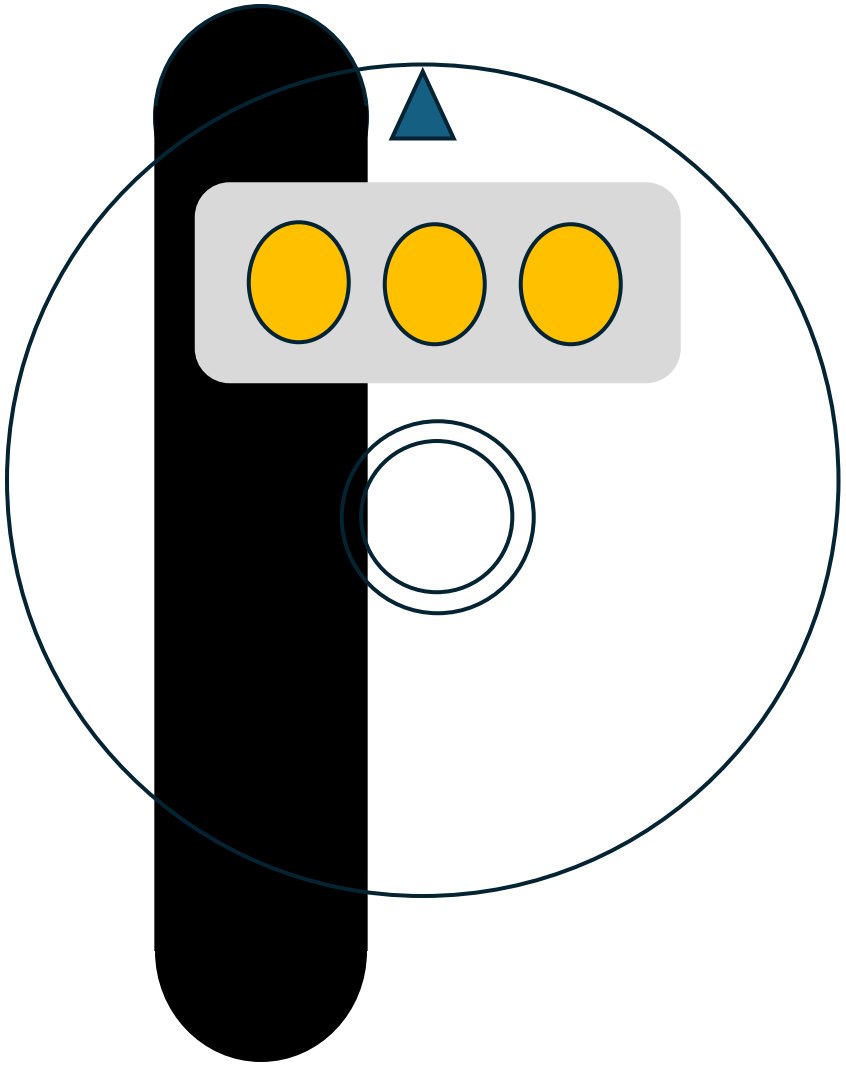
- `tuple`: (right_line, center_line, left_line) - Right, center, left line detection values

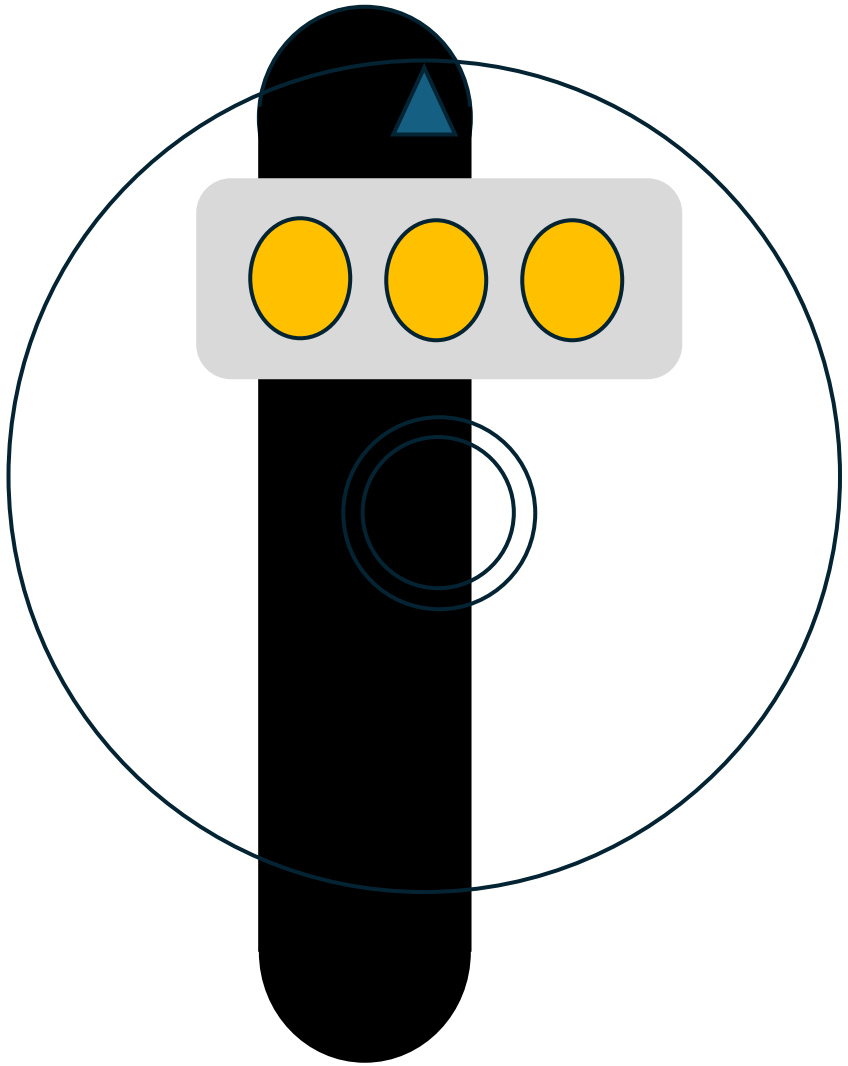
Example:

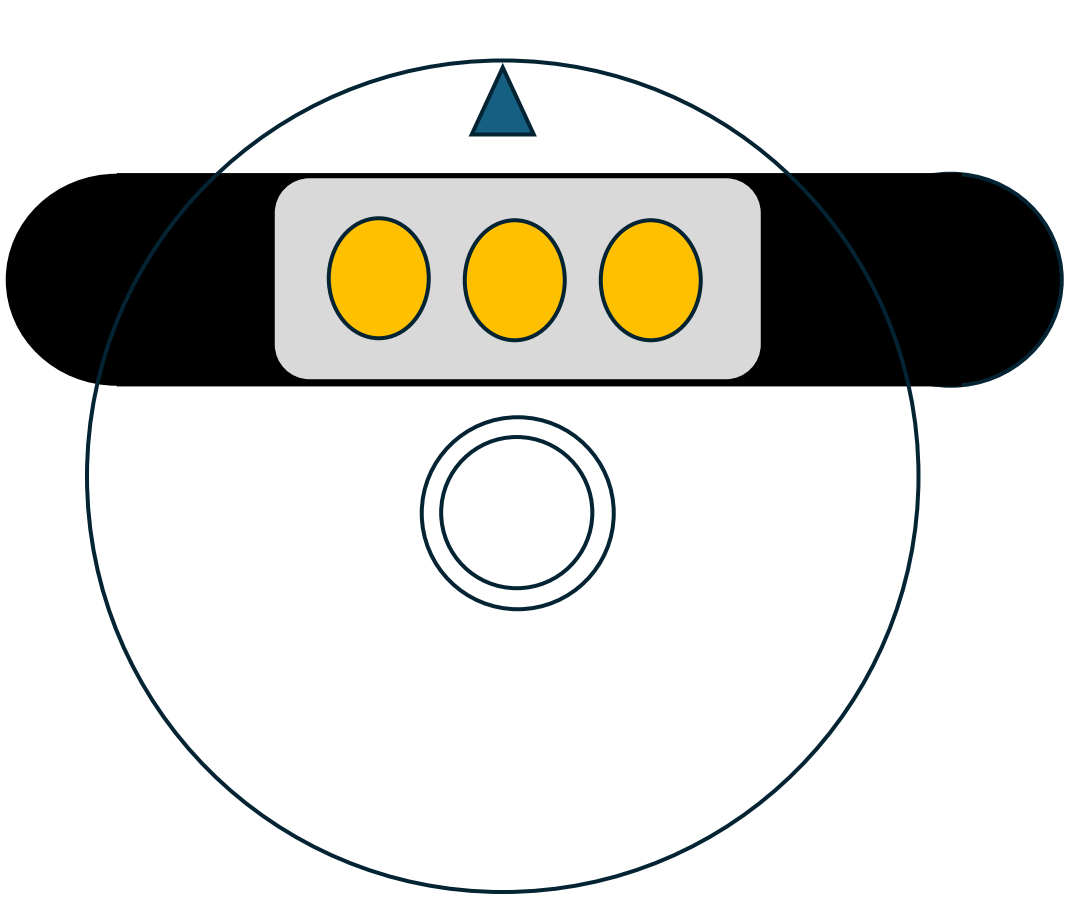
```python
right, center, left = robot.get_line_sensor(True)
print(f"Right: {right}, Center: {center}, Left: {left}")
```

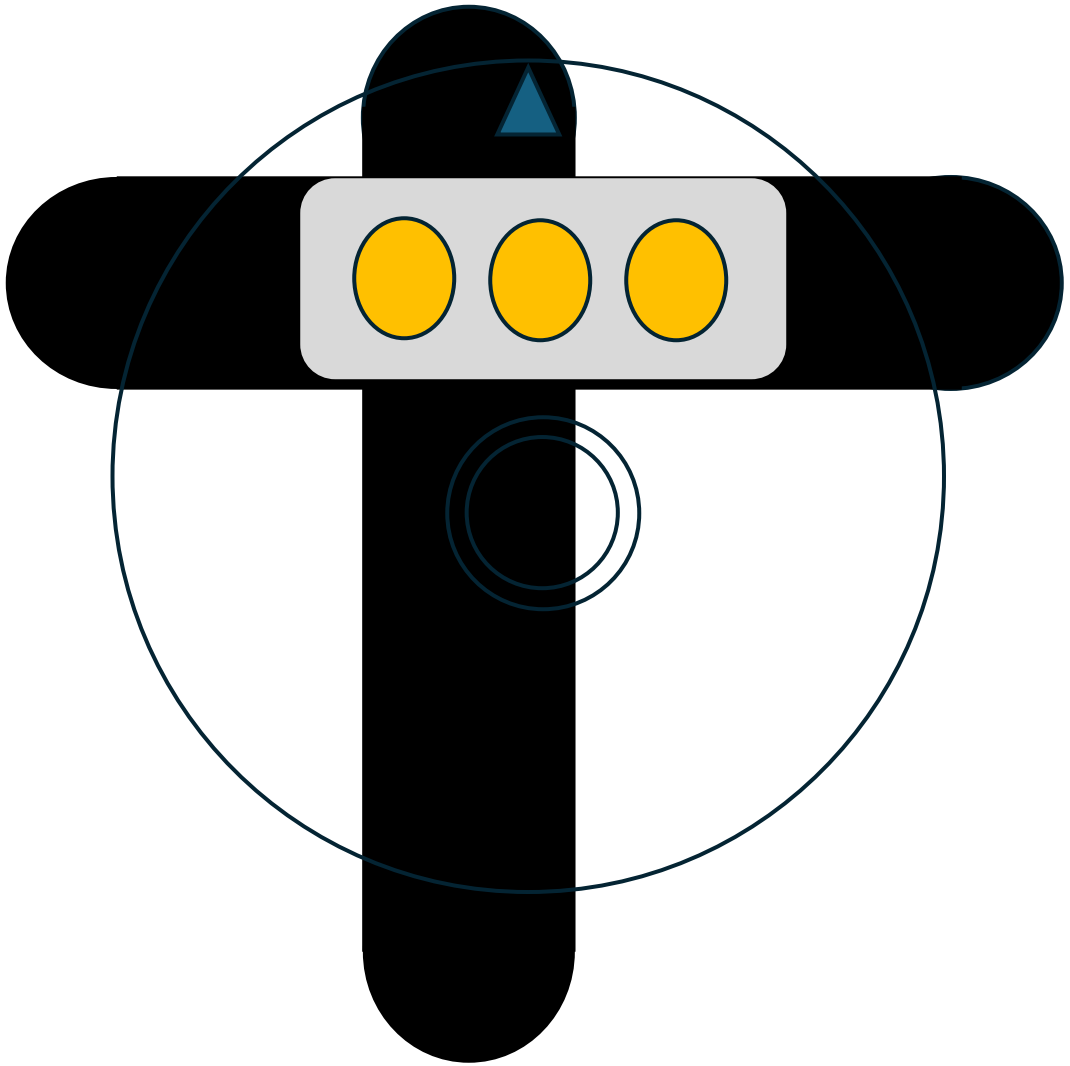
Black: 1, Others: 0


File Edit Selection View ···
pi_manual
main.py U
samples > main.py > ...
robot = KamibotPi('COM8', 57600)
robot.init()
try:
# YOUR CODE
right, center, left = robot.get_line_sensor(True)
print(f"Right: {right}, Center: {center}, Left: {left}")
except KeyboardInterrupt:
print("Program stopped by user")
finally:
robot.stop()
robot.close()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
(pi) PS D:\wk_scratch\pi_manual>
main* 0 0 Ln 15, Col 17 Spaces: 2 UTF-8 CRLF Python 3.10.0 (pi) Go Live









Color Sensor


Color
Sensor

`get_color_elements(opt)`

Activates the color sensor and reads RGB values.

Parameters:

- `opt` (bool, optional): Sensor operation mode
  - `True`: Activate sensor (default)
  - `False`: Stop sensor

Returns:

- `tuple`: (r, g, b) - Red, Green, Blue values

Example:

```python
r, g, b = robot.get_color_elements(True)
print(f"RGB: ({r}, {g}, {b})")
```


File Edit Selection View ···
pi_manual
main.py U
samples > main.py > ...
robot = KamibotPi('COM8', 57600)
robot.init()
try:
# YOUR CODE
r, g, b = robot.get_color_elements(True)
print(f"RGB: ({r}, {g}, {b})")
except KeyboardInterrupt:
print("Program stopped by user")
finally:
robot.stop()
robot.close()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
(pi) PS D:\wk_scratch\pi_manual>
main* 0 0 Ln 9, Col 5 Spaces: 2 UTF-8 CRLF Python 3.10.0 (pi) Go Live

`diff_color(color1, color2)` (Global Function)

Calculates the Euclidean distance between two RGB colors.

Parameters:

- `color1` (tuple): First color (r, g, b)
- `color2` (tuple): Second color (r, g, b)

Returns:

- `float`: Distance between colors (closer to 0 means more similar)

Example:

```python
from kamibotpi import diff_color

color1 = (255, 0, 0)  # Red
color2 = (255, 100, 0)  # Orange
distance = diff_color(color1, color2)
print(f"Color difference: {distance}")
```

To determine the color from the measurement values of a color sensor, one can judge by calculating the distance to each color.

File Edit Selection View ··· pi_manual

main.py U

samples > main.py > ...

```
    # YOUR CODE
    RED = (255, 0, 0)
    BLUE = (0, 0, 255)
    r, g, b = robot.get_color_elements(True)
    print(f"RGB: ({r}, {g}, {b})")

    print(robot.diff_color(RED, (r, g, b)))
    print(robot.diff_color(BLUE, (r, g, b)))

except KeyboardInterrupt:
    print("Program stopped by user")

finally:
    robot.stop()
    robot.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python

```
(pi) PS D:\wk_scratch\pi_manual>
```

main* 0 0 Ln 17, Col 37 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live

Make beep

`beep()`

Makes a short beep sound.

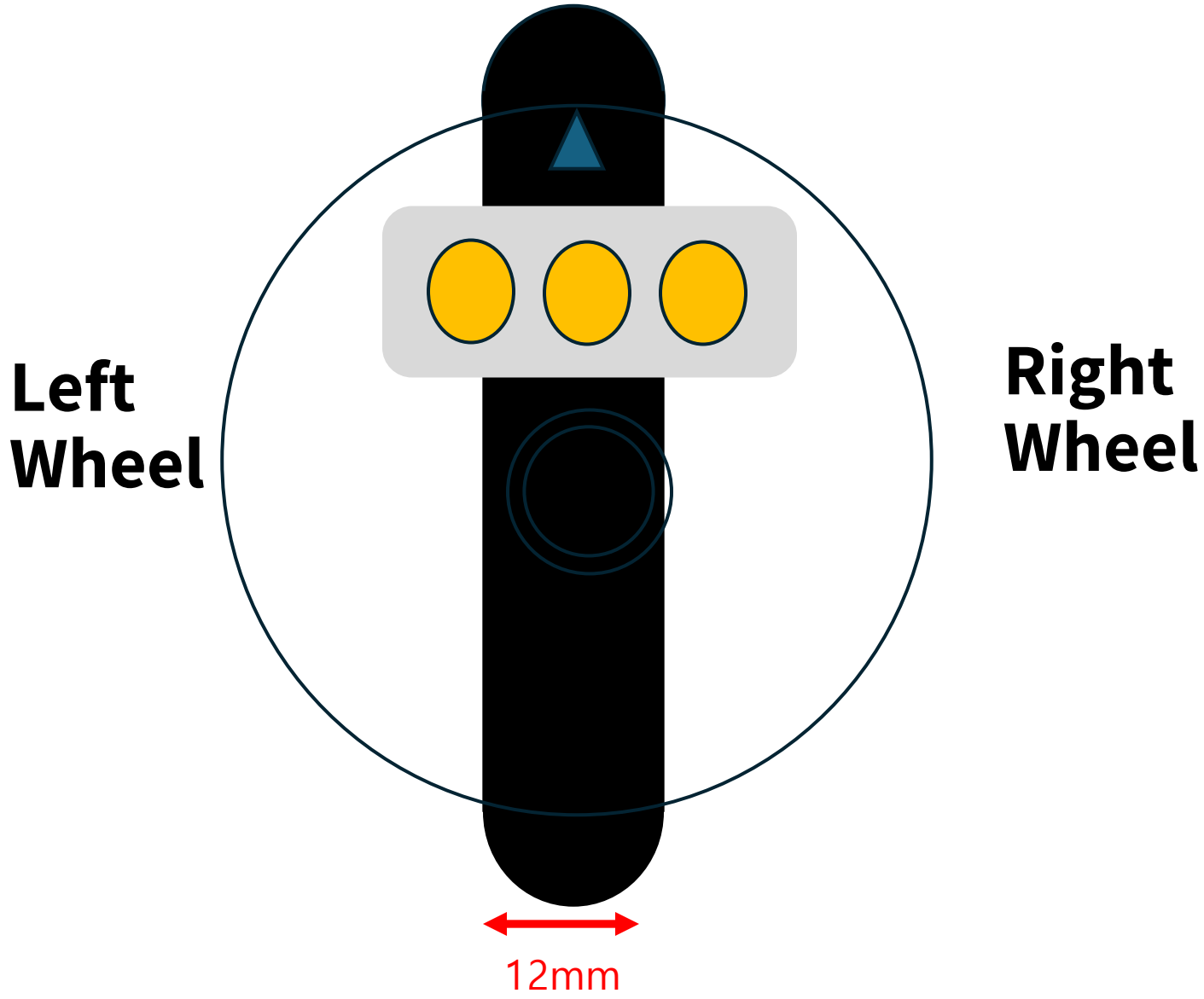
Parameters: None

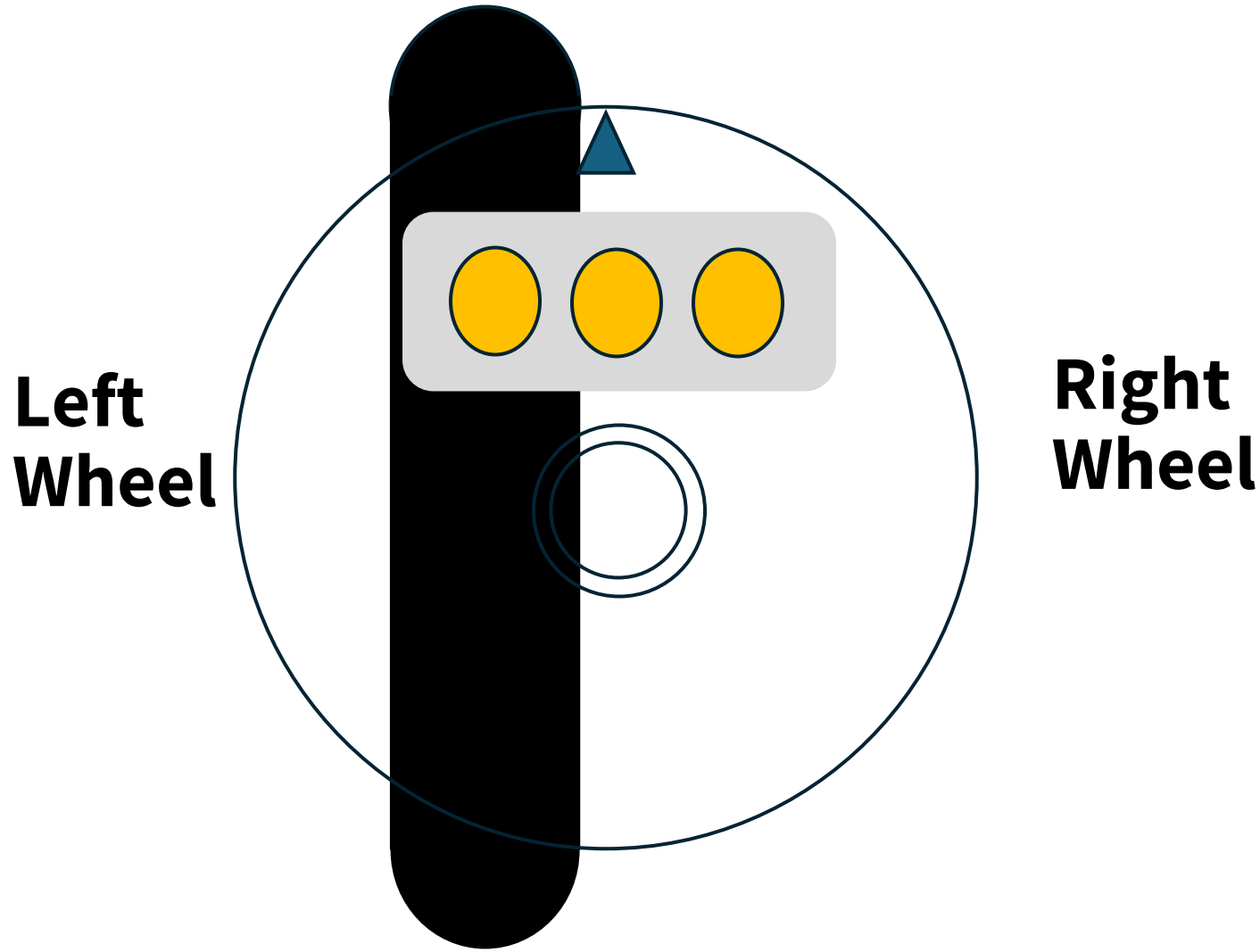
Returns: None

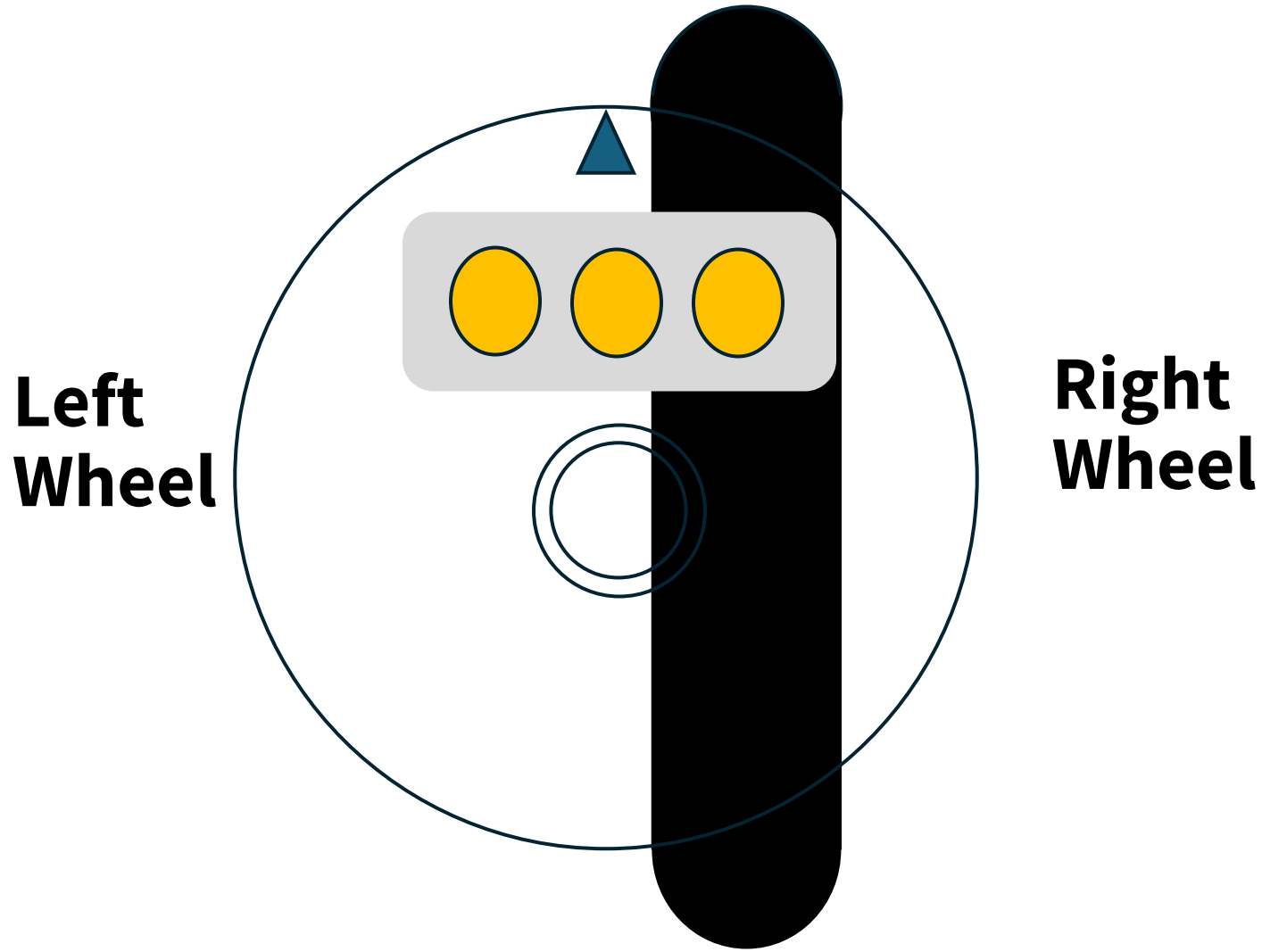
Example:

```python
robot.beep()  # Short beep
```

The Principle of the Line Follower Robot


Left
Wheel
Right
Wheel
12mm


Left
Wheel
Right
Wheel


Left
Wheel
Right
Wheel

File Edit Selection View ··· pi_manual

main.py U

samples > main.py > ...

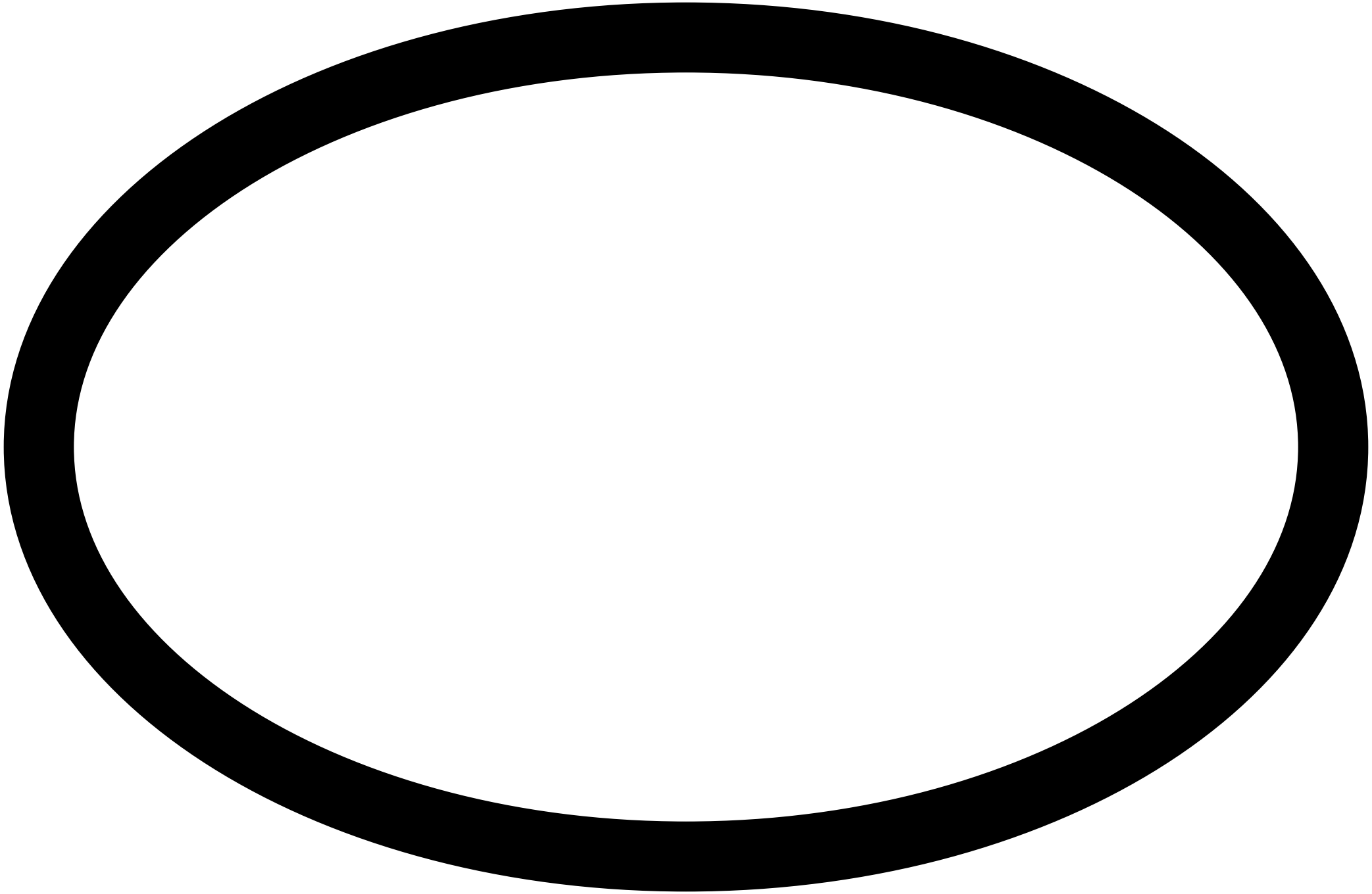
```python
robot.init()

try:
    # YOUR CODE
    right, center, left = robot.get_line_sensor(True)
    print(f"Right: {right}, Center: {center}, Left: {left}")

    if center == 1:
        robot.go_dir_speed("f", 50, "f", 50)
    elif left == 1:
        pass
    elif right == 1:
        pass

except KeyboardInterrupt:
```

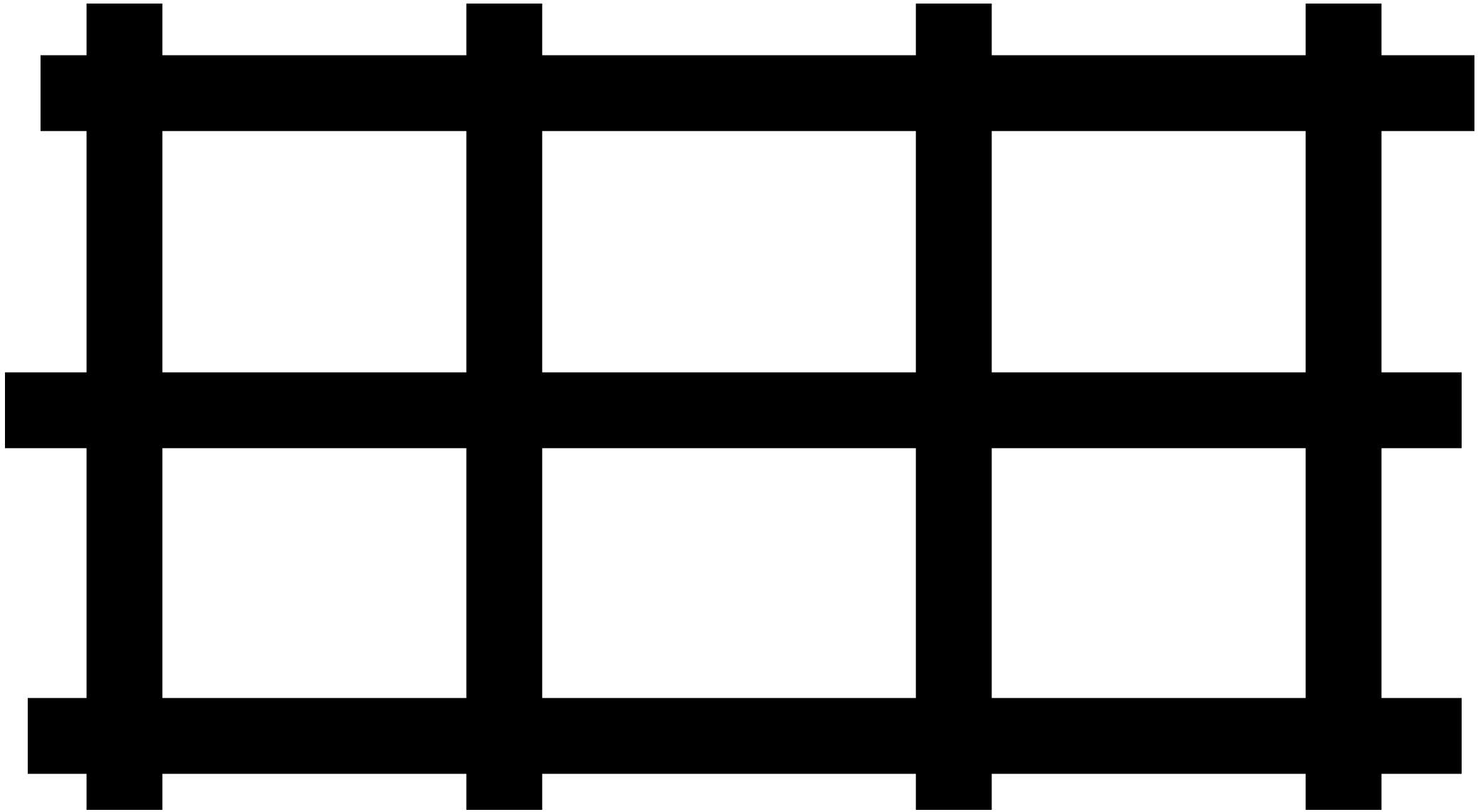
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python

```
(pi) PS D:\wk_scratch\pi_manual>
```

main* 0 0 Ln 14, Col 13 Spaces: 2 UTF-8 CRLF {} Python 3.10.0 (pi) Go Live





END