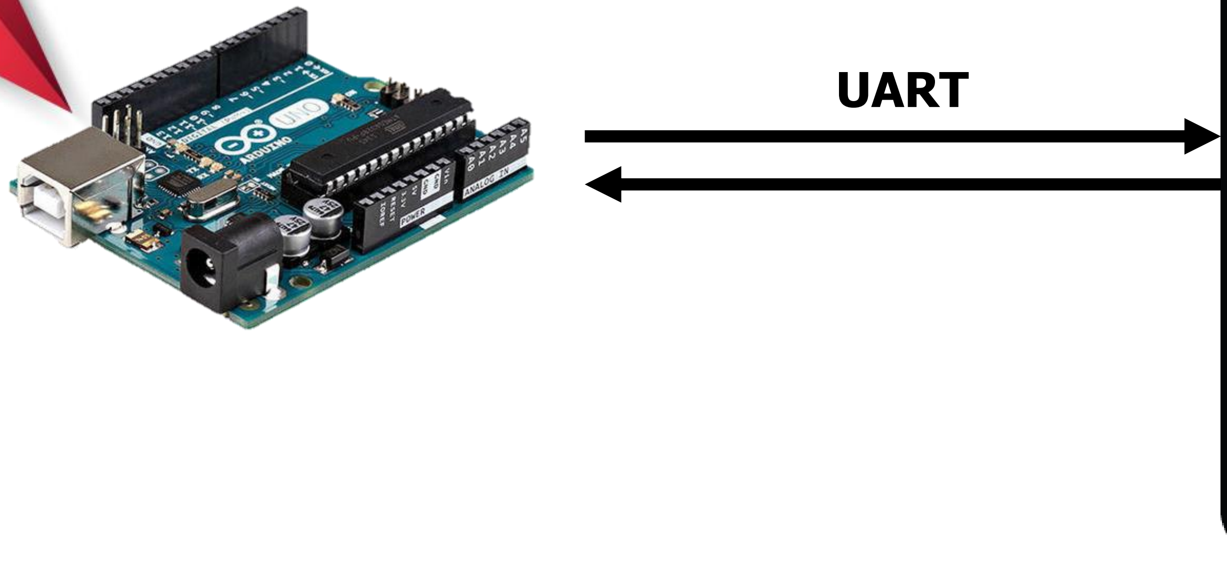
Python

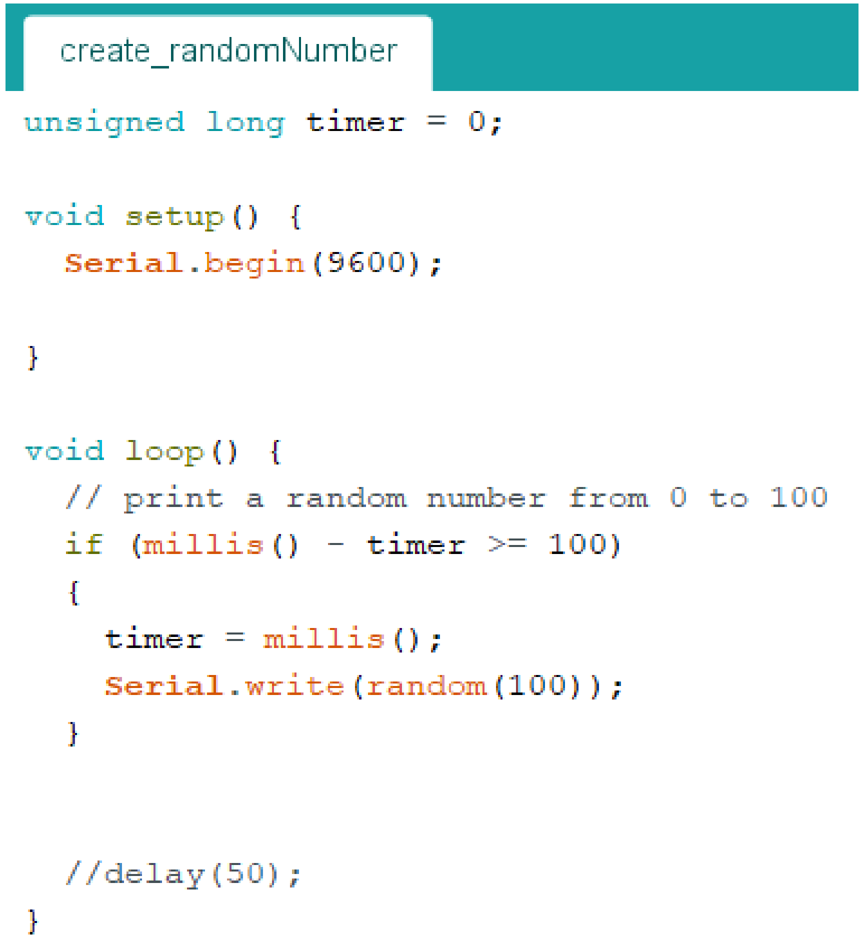
Thread Serial Pyqt

CuBoBKDN
bo.phamconganhhuy@gmail.com
tapit.vn


UART
Qt

Arduino Serial

Gửi giá trị ngẫu nhiên qua Serial

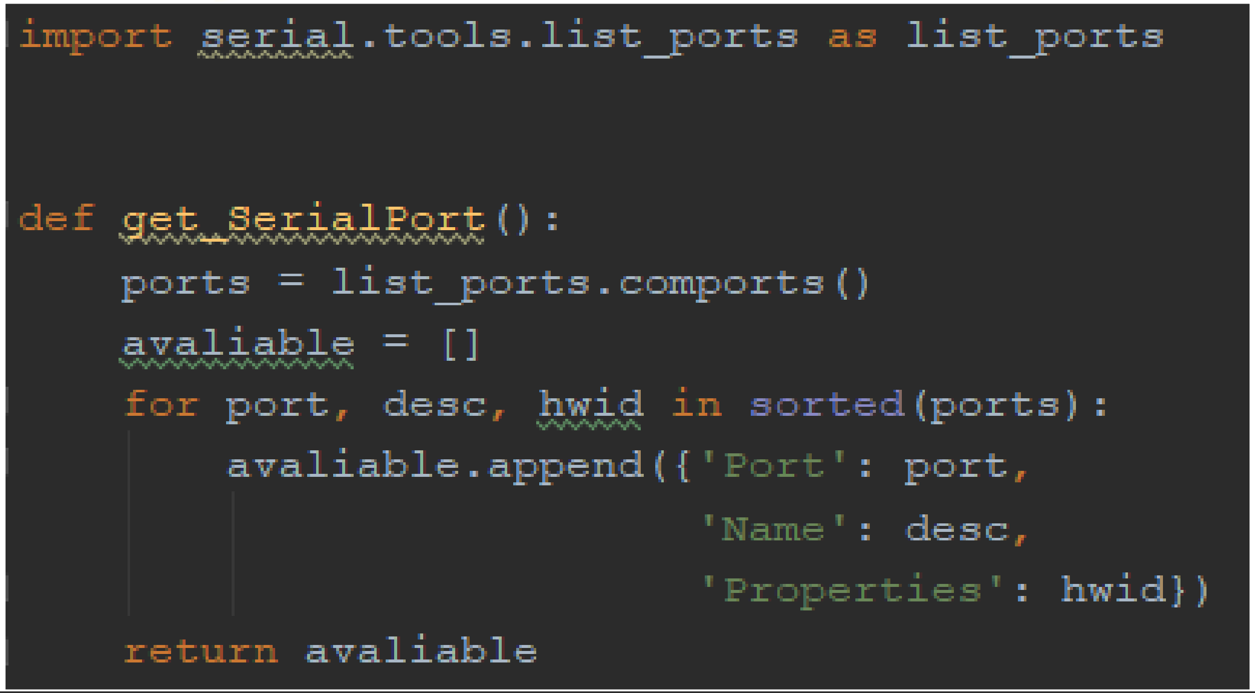


```
unsigned long timer = 0;

void setup() {
  Serial.begin(9600);

}

void loop() {
  // print a random number from 0 to 100
  if (millis() - timer >= 100)
  {
    timer = millis();
    Serial.write(random(100));
  }


  //delay(50);
}
```

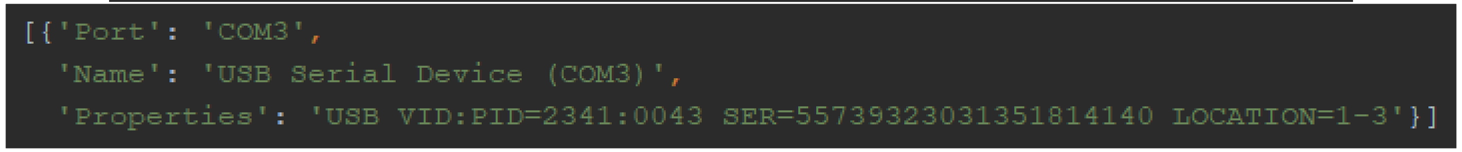
serial-tool\arduino\create_randomNumber\create_randomNumber.ino

Pyserial
lấy danh sách Serial Port

Yêu cầu
Thư viện pyserial

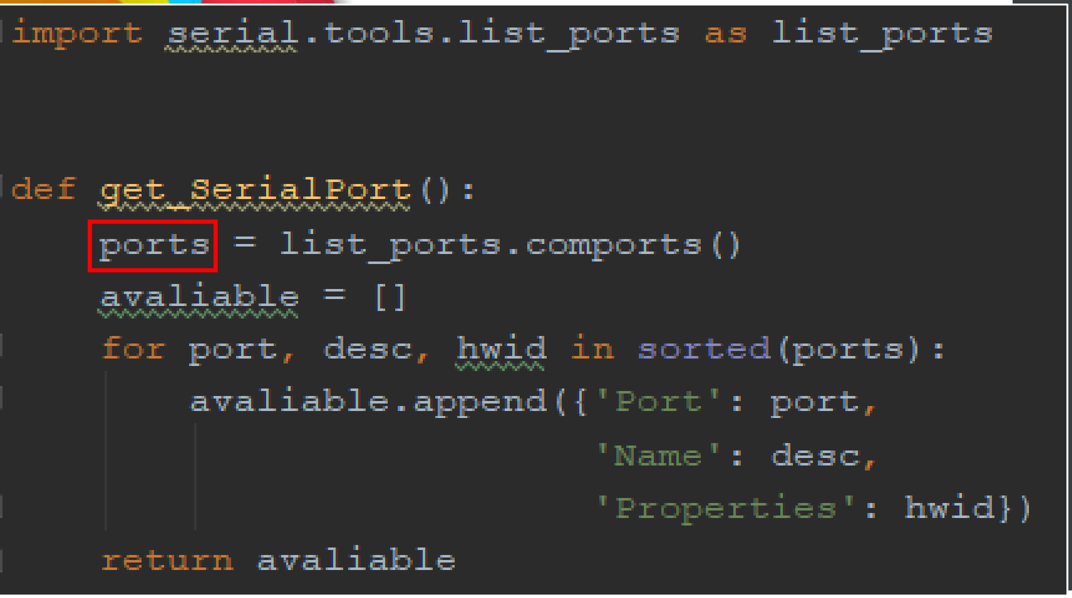


```python
import serial.tools.list_ports as list_ports


def get_SerialPort():
    ports = list_ports.comports()
    avaliable = []
    for port, desc, hwid in sorted(ports):
        avaliable.append({'Port': port,
                          'Name': desc,
                          'Properties': hwid})
    return avaliable
```

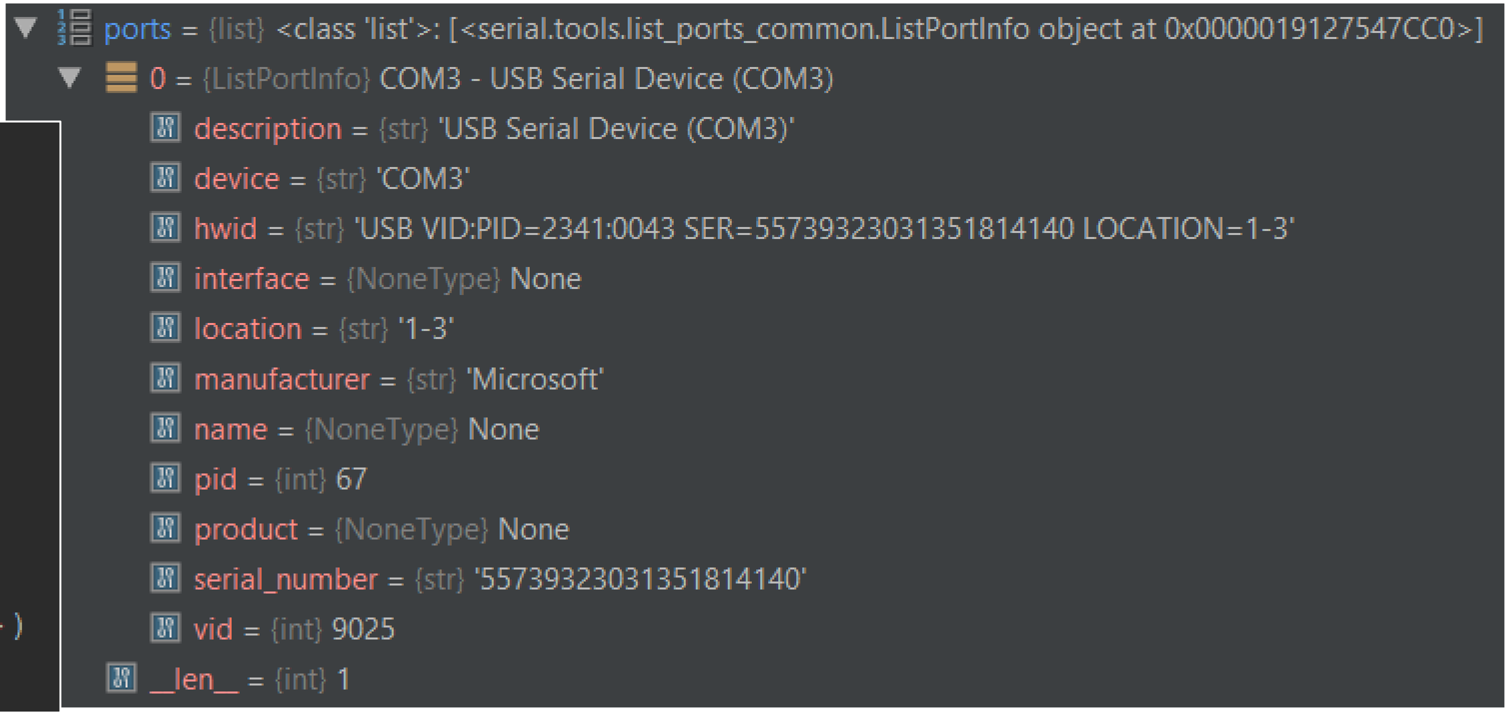


```
[{'Port': 'COM3',
  'Name': 'USB Serial Device (COM3)',
  'Properties': 'USB VID:PID=2341:0043 SER=55739323031351814140 LOCATION=1-3'}]
```

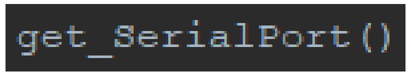
Pyserial
lấy danh sách Serial Port



```python
import serial.tools.list_ports as list_ports


def get_SerialPort():
    ports = list_ports.comports()
    avaliable = []
    for port, desc, hwid in sorted(ports):
        avaliable.append({'Port': port,
                          'Name': desc,
                          'Properties': hwid})
    return avaliable
```

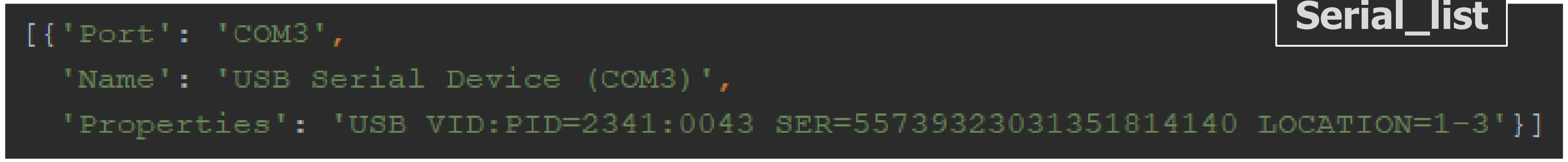


```
▼ ports = {list} <class 'list'>: [<serial.tools.list_ports_common.ListPortInfo object at 0x0000019127547CC0>]
   ▼ 0 = {ListPortInfo} COM3 - USB Serial Device (COM3)
        description = {str} 'USB Serial Device (COM3)'
        device = {str} 'COM3'
        hwid = {str} 'USB VID:PID=2341:0043 SER=55739323031351814140 LOCATION=1-3'
        interface = {NoneType} None
        location = {str} '1-3'
        manufacturer = {str} 'Microsoft'
        name = {NoneType} None
        pid = {int} 67
        product = {NoneType} None
        serial_number = {str} '55739323031351814140'
        vid = {int} 9025
     __len__ = {int} 1
```



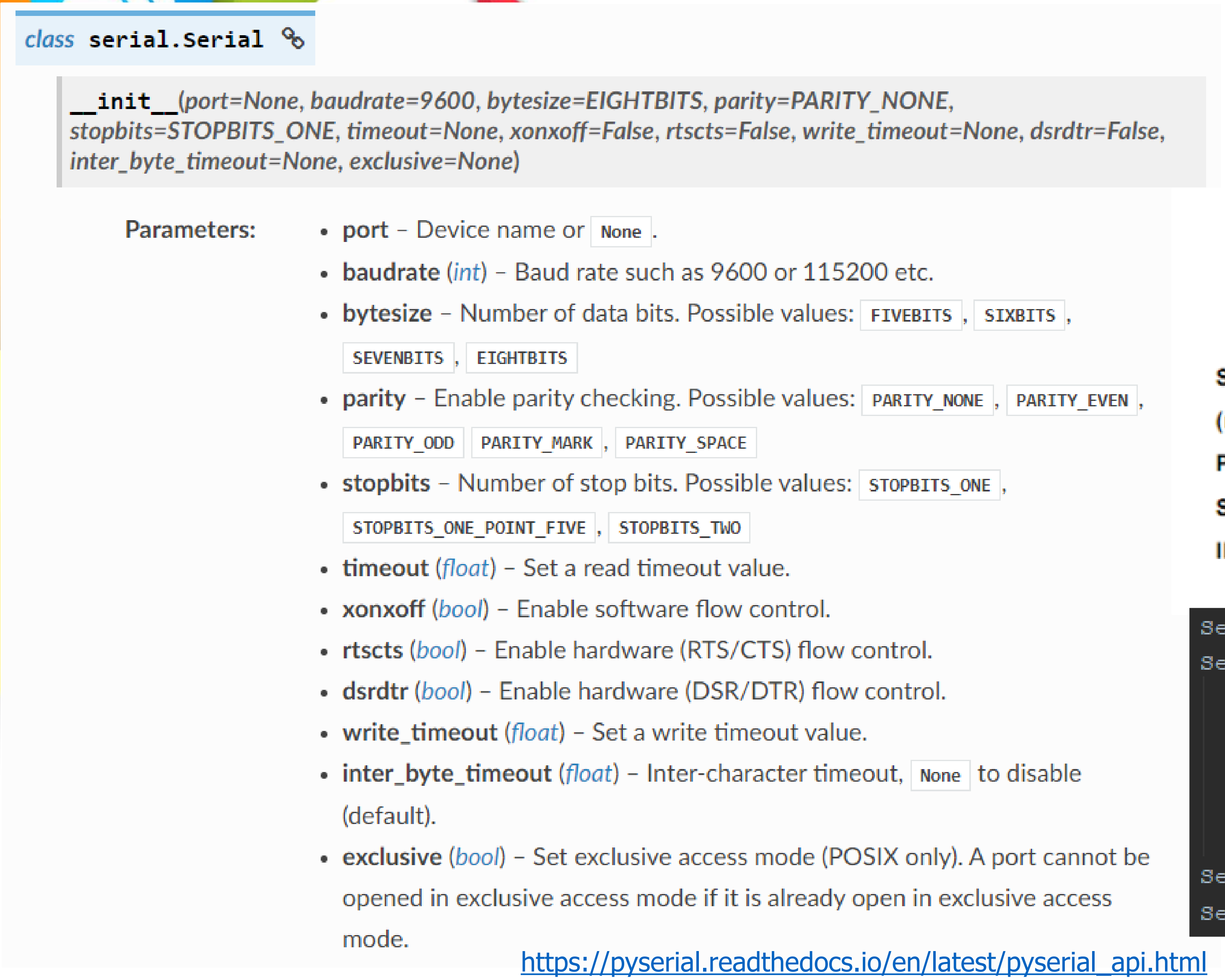
```
get_SerialPort()
```

Serial_list



```
[{'Port': 'COM3',
  'Name': 'USB Serial Device (COM3)',
  'Properties': 'USB VID:PID=2341:0043 SER=55739323031351814140 LOCATION=1-3'}]
```

Pyserial
Tạo kết nối Serial

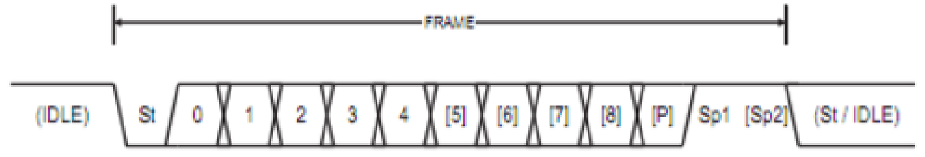


class **serial.Serial**

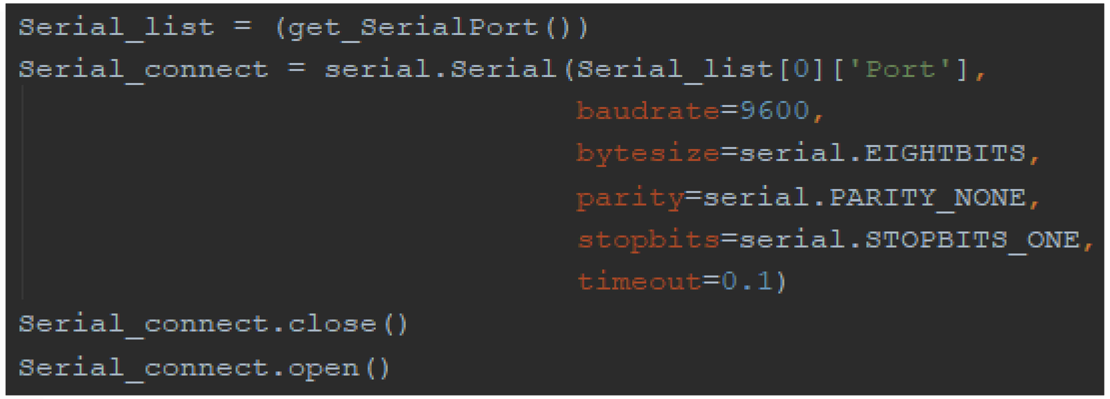
__init__(*port=None, baudrate=9600, bytesize=EIGHTBITS, parity=PARITY_NONE, stopbits=STOPBITS_ONE, timeout=None, xonxoff=False, rtscts=False, write_timeout=None, dsrdtr=False, inter_byte_timeout=None, exclusive=None*)

Parameters:

- **port** – Device name or `None`.
- **baudrate** (*int*) – Baud rate such as 9600 or 115200 etc.
- **bytesize** – Number of data bits. Possible values: `FIVEBITS`, `SIXBITS`, `SEVENBITS`, `EIGHTBITS`
- **parity** – Enable parity checking. Possible values: `PARITY_NONE`, `PARITY_EVEN`, `PARITY_ODD` `PARITY_MARK`, `PARITY_SPACE`
- **stopbits** – Number of stop bits. Possible values: `STOPBITS_ONE`, `STOPBITS_ONE_POINT_FIVE`, `STOPBITS_TWO`
- **timeout** (*float*) – Set a read timeout value.
- **xonxoff** (*bool*) – Enable software flow control.
- **rtscts** (*bool*) – Enable hardware (RTS/CTS) flow control.
- **dsrdtr** (*bool*) – Enable hardware (DSR/DTR) flow control.
- **write_timeout** (*float*) – Set a write timeout value.
- **inter_byte_timeout** (*float*) – Inter-character timeout, `None` to disable (default).
- **exclusive** (*bool*) – Set exclusive access mode (POSIX only). A port cannot be opened in exclusive access mode if it is already open in exclusive access mode.



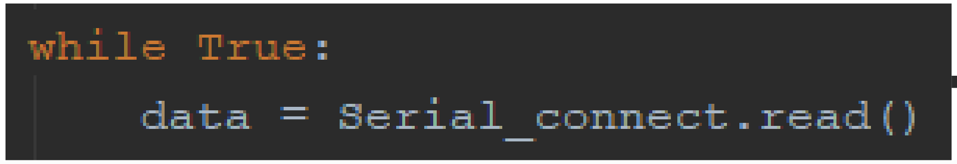
St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxDn or TxDn). An IDLE line must be high.



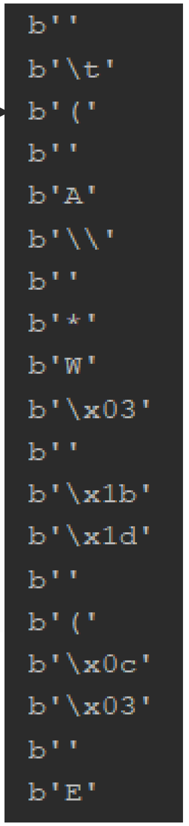
```
Serial_list = (get_SerialPort())
Serial_connect = serial.Serial(Serial_list[0]['Port'],
                               baudrate=9600,
                               bytesize=serial.EIGHTBITS,
                               parity=serial.PARITY_NONE,
                               stopbits=serial.STOPBITS_ONE,
                               timeout=0.1)
Serial_connect.close()
Serial_connect.open()
```

https://pyserial.readthedocs.io/en/latest/pyserial_api.html

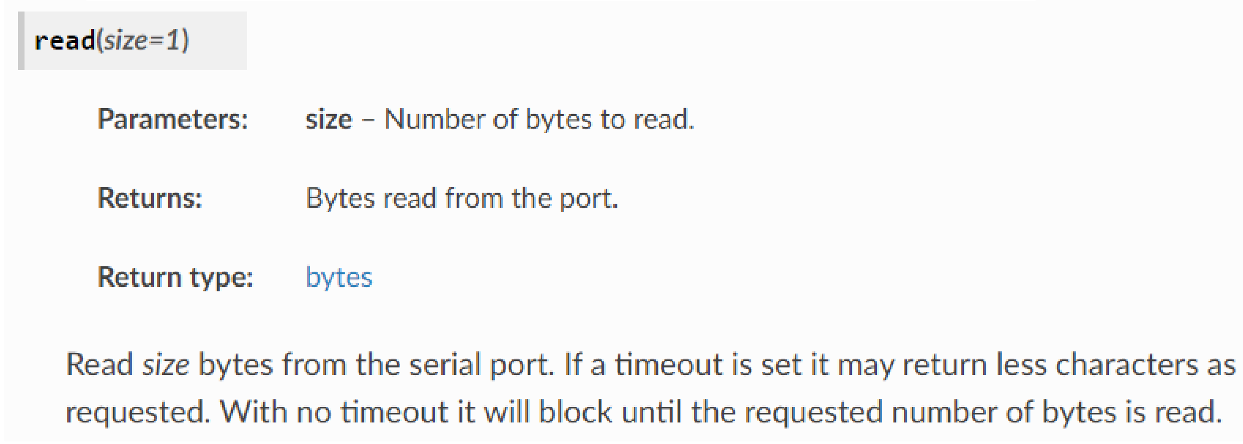
Pyserial
Nhận giá trị từ Serial



```
while True:
    data = Serial_connect.read()
```



```
b''
b'\t'
b'('
b''
b'A'
b'\\'
b''
b'*'
b'W'
b'\x03'
b''
b'\x1b'
b'\x1d'
b''
b'('
b'\x0c'
b'\x03'
b''
b'E'
```



read(*size=1*)

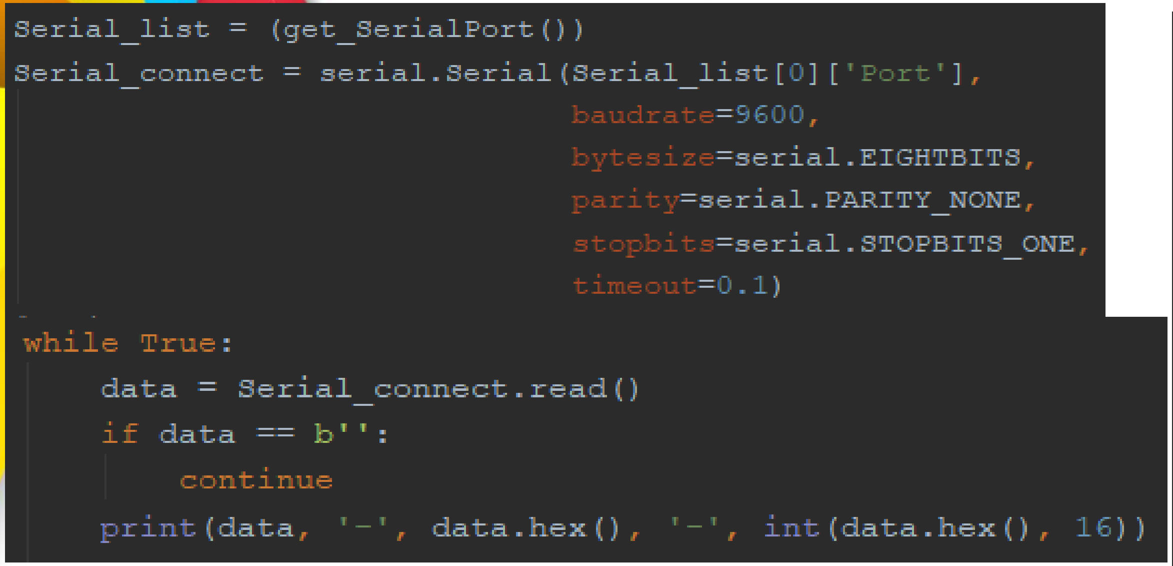
Parameters:	**size** – Number of bytes to read.
Returns:	Bytes read from the port.
Return type:	bytes

Read *size* bytes from the serial port. If a timeout is set it may return less characters as requested. With no timeout it will block until the requested number of bytes is read.

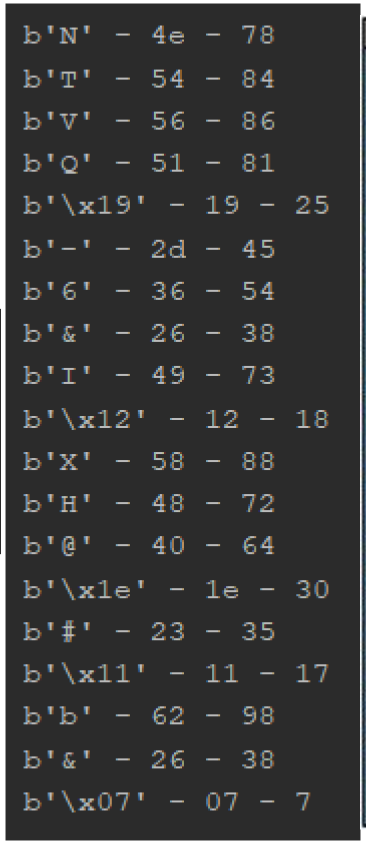
https://pyserial.readthedocs.io/en/latest/pyserial_api.html

Pyserial

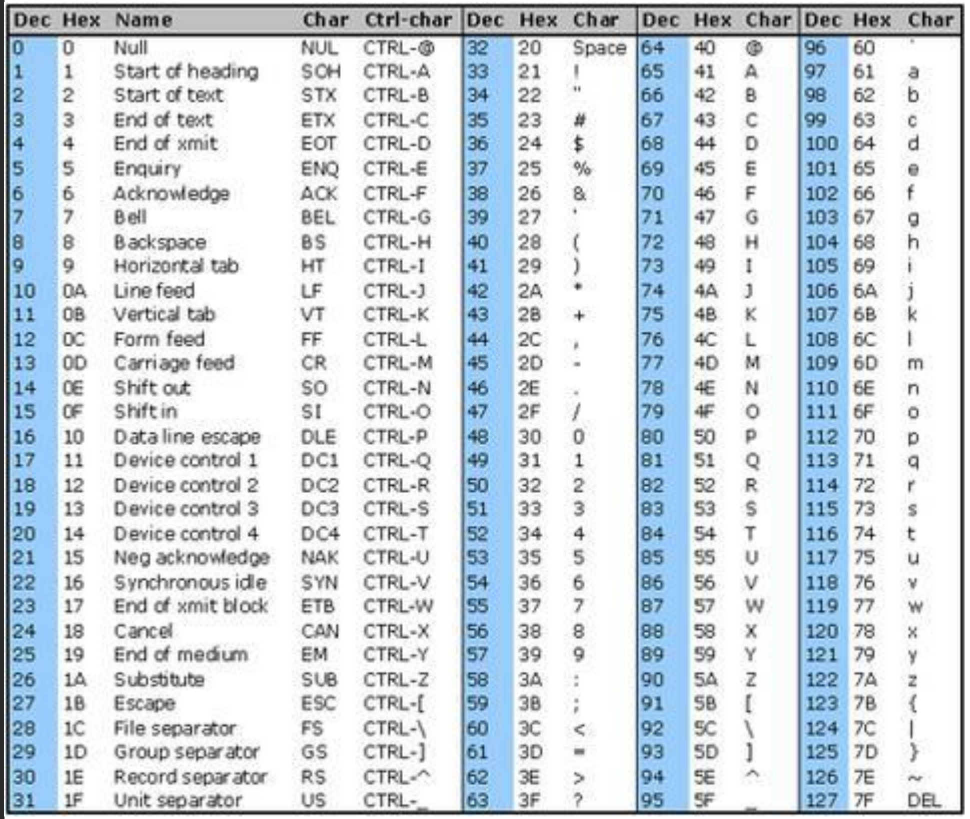
Nhận giá trị từ Serial



```python
Serial_list = (get_SerialPort())
Serial_connect = serial.Serial(Serial_list[0]['Port'],
                               baudrate=9600,
                               bytesize=serial.EIGHTBITS,
                               parity=serial.PARITY_NONE,
                               stopbits=serial.STOPBITS_ONE,
                               timeout=0.1)

while True:
    data = Serial_connect.read()
    if data == b'':
        continue
    print(data, '-', data.hex(), '-', int(data.hex(), 16))
```

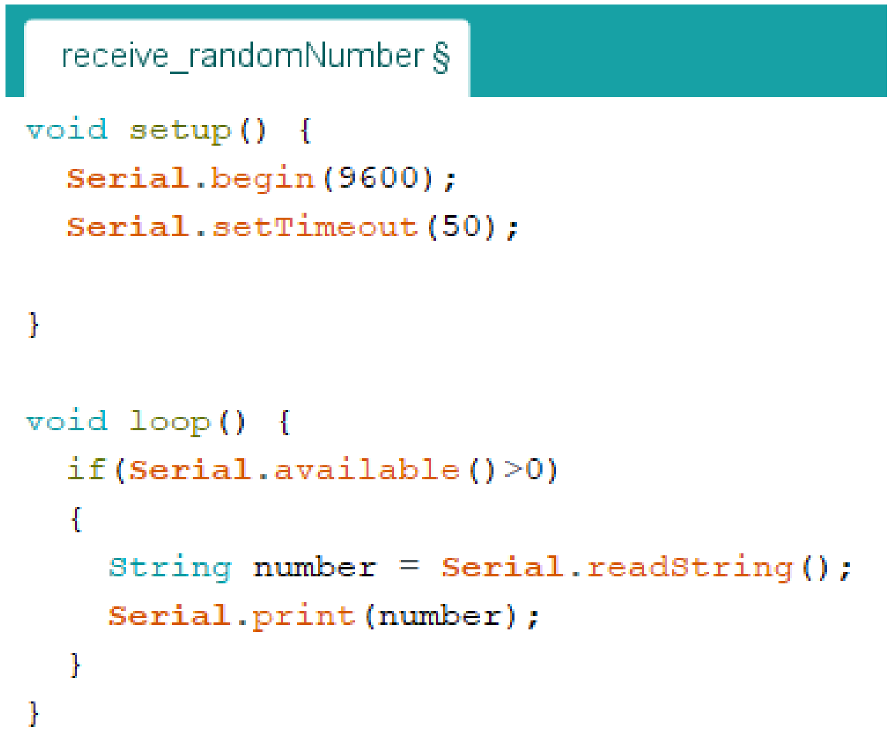


```
b'N' - 4e - 78
b'T' - 54 - 84
b'V' - 56 - 86
b'Q' - 51 - 81
b'\x19' - 19 - 25
b'-' - 2d - 45
b'6' - 36 - 54
b'&' - 26 - 38
b'I' - 49 - 73
b'\x12' - 12 - 18
b'X' - 58 - 88
b'H' - 48 - 72
b'@' - 40 - 64
b'\x1e' - 1e - 30
b'#' - 23 - 35
b'\x11' - 11 - 17
b'b' - 62 - 98
b'&' - 26 - 38
b'\x07' - 07 - 7
```



Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Arduino Serial
Nhận và trả về giá trị

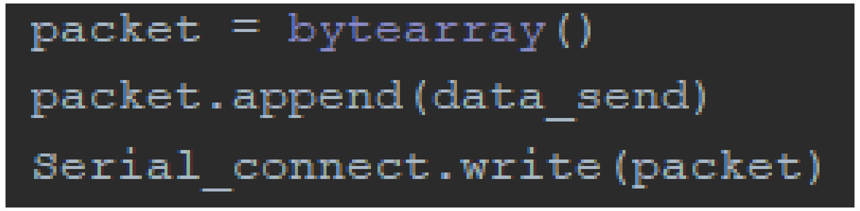

receive_randomNumber §

```
void setup() {
  Serial.begin(9600);
  Serial.setTimeout(50);

}

void loop() {
  if(Serial.available()>0)
  {
    String number = Serial.readString();
    Serial.print(number);
  }
}
```

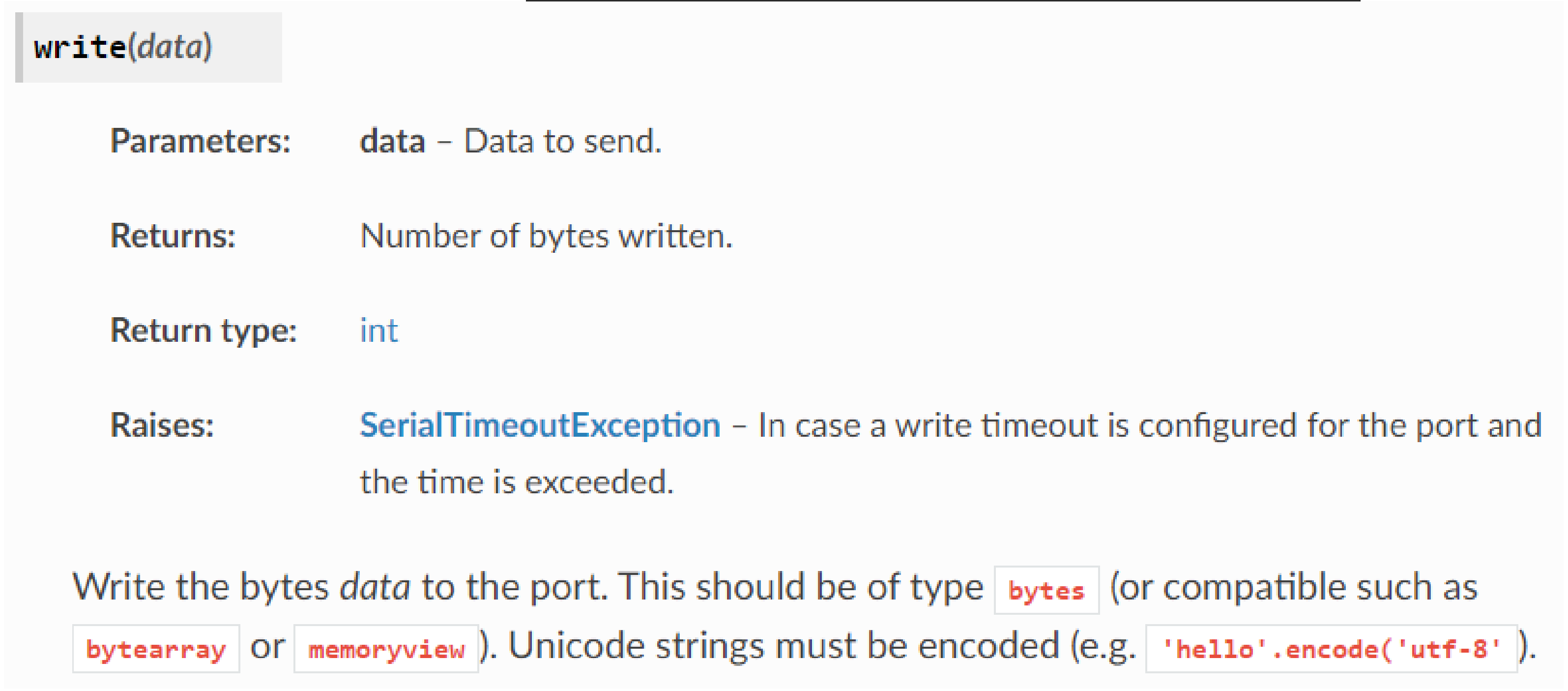
serial-tool\arduino\receive_randomNumber\receive_randomNumber.ino

Pyserial

Gửi giá trị qua Serial



```
packet = bytearray()
packet.append(data_send)
Serial_connect.write(packet)
```



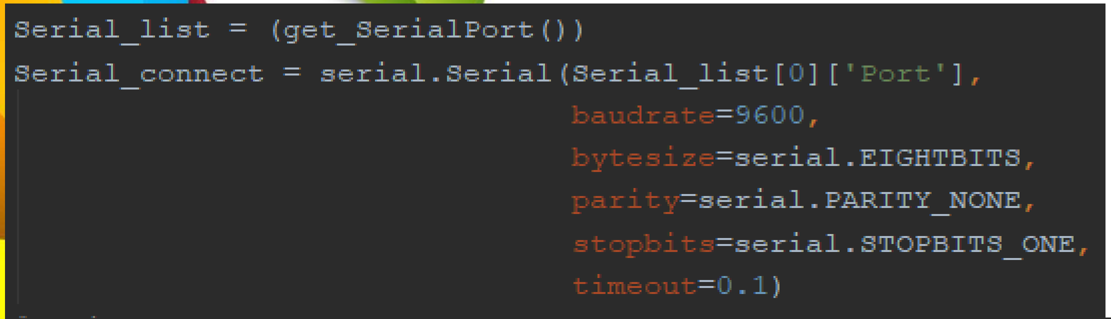
`write(data)`

Parameters:	**data** – Data to send.
Returns:	Number of bytes written.
Return type:	int
Raises:	SerialTimeoutException – In case a write timeout is configured for the port and the time is exceeded.

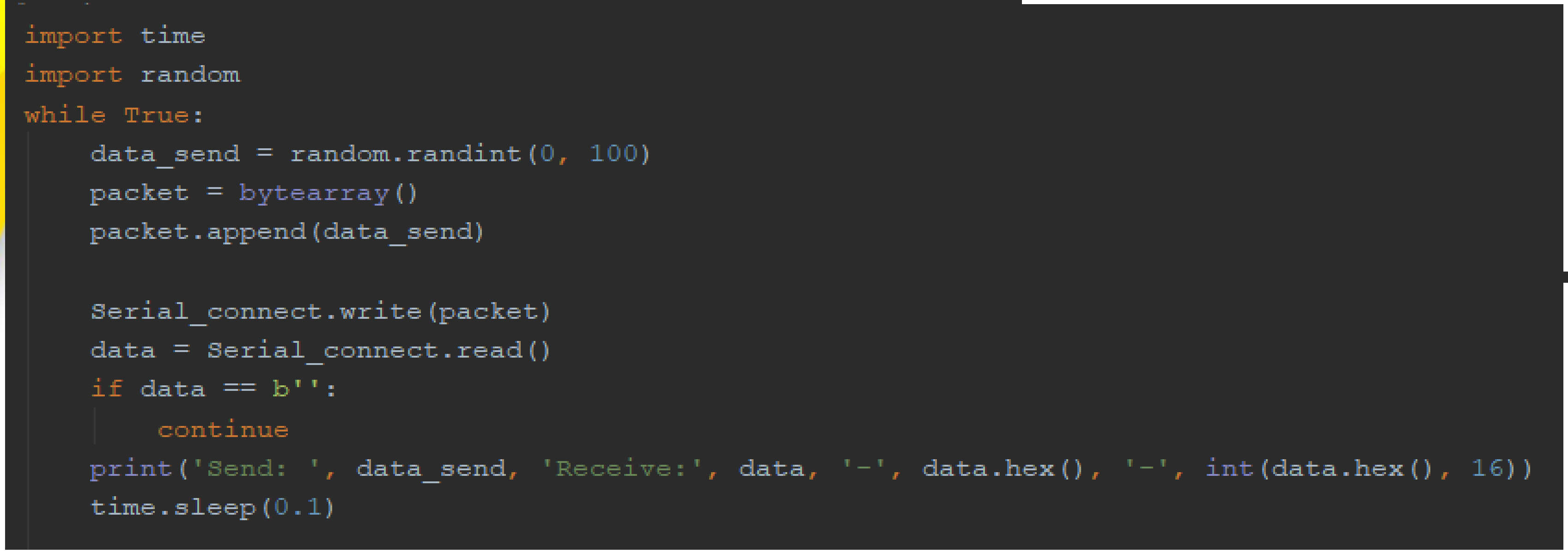
Write the bytes *data* to the port. This should be of type `bytes` (or compatible such as `bytearray` or `memoryview`). Unicode strings must be encoded (e.g. `'hello'.encode('utf-8'`).

https://pyserial.readthedocs.io/en/latest/pyserial_api.html

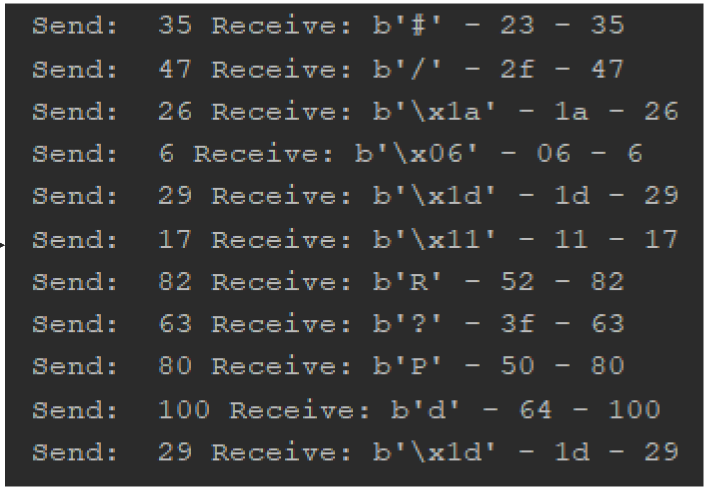
Pyserial
Gửi giá trị qua Serial



```
Serial_list = (get_SerialPort())
Serial_connect = serial.Serial(Serial_list[0]['Port'],
                               baudrate=9600,
                               bytesize=serial.EIGHTBITS,
                               parity=serial.PARITY_NONE,
                               stopbits=serial.STOPBITS_ONE,
                               timeout=0.1)
```



```
import time
import random
while True:
    data_send = random.randint(0, 100)
    packet = bytearray()
    packet.append(data_send)

    Serial_connect.write(packet)
    data = Serial_connect.read()
    if data == b'':
        continue
    print('Send: ', data_send, 'Receive:', data, '-', data.hex(), '-', int(data.hex(), 16))
    time.sleep(0.1)
```



```
Send:  35 Receive: b'#' - 23 - 35
Send:  47 Receive: b'/' - 2f - 47
Send:  26 Receive: b'\x1a' - 1a - 26
Send:  6 Receive: b'\x06' - 06 - 6
Send:  29 Receive: b'\x1d' - 1d - 29
Send:  17 Receive: b'\x11' - 11 - 17
Send:  82 Receive: b'R' - 52 - 82
Send:  63 Receive: b'?' - 3f - 63
Send:  80 Receive: b'P' - 50 - 80
Send:  100 Receive: b'd' - 64 - 100
Send:  29 Receive: b'\x1d' - 1d - 29
```

Pyserial vs Arduino Serial
Problem

Arduino Serial 100byte/s $\longrightarrow$ **Pyserial** 100byte/s ≡ 10ms/byte

+

Chương trình xử lý

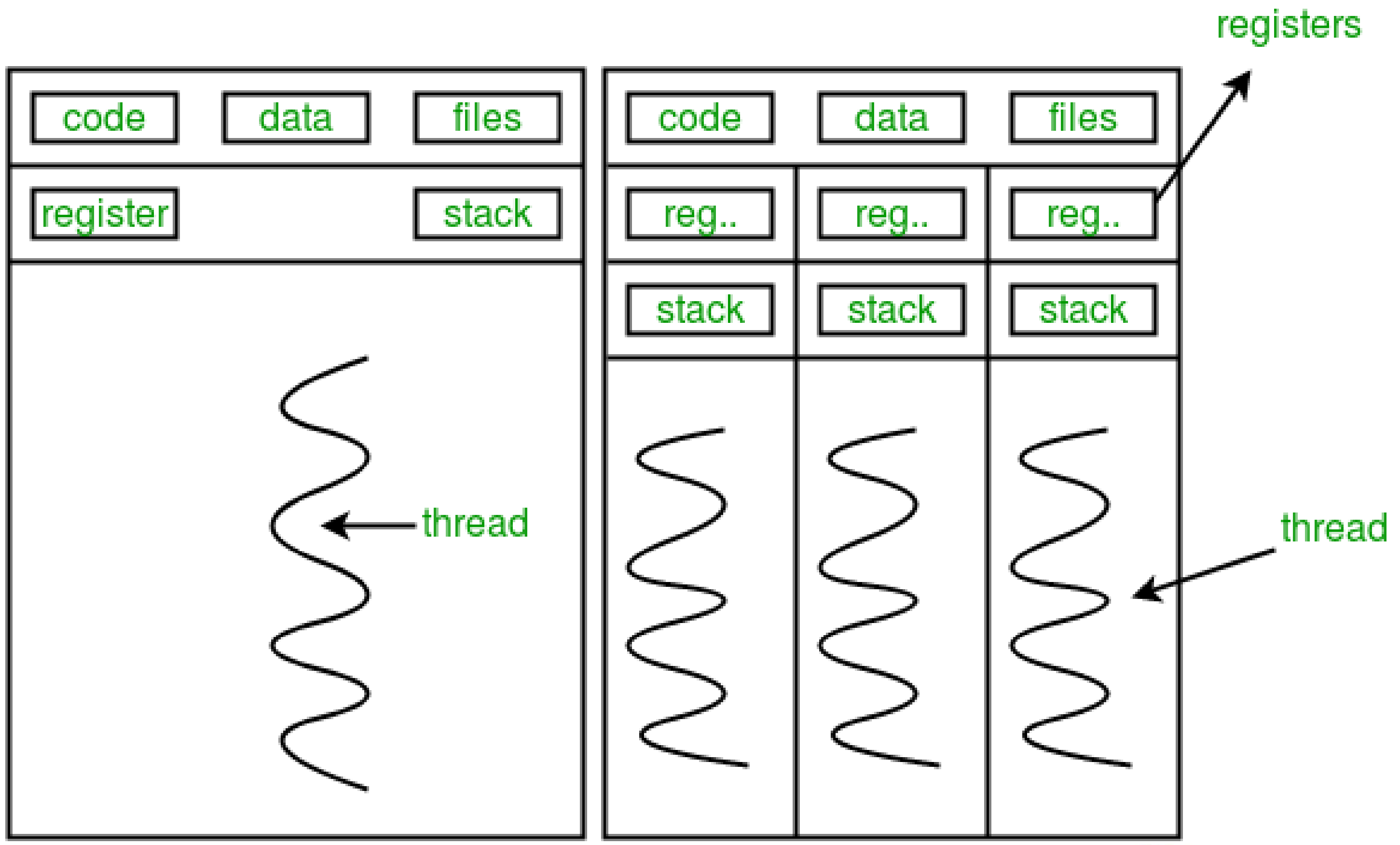
$\downarrow$

Không nhận kịp thời data từ Arduino Serial

$\downarrow$

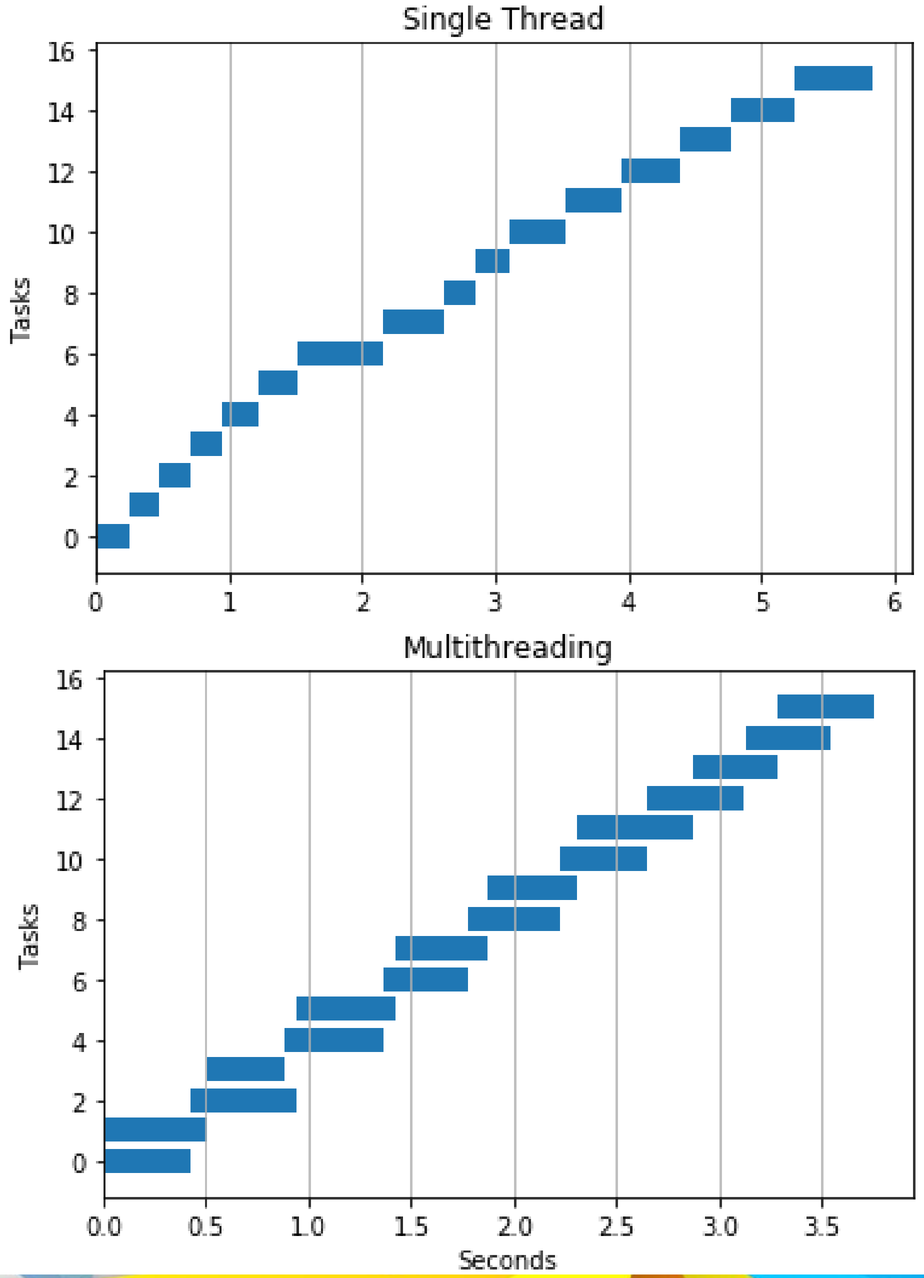
Không thể xử lý tuần tự

Thread

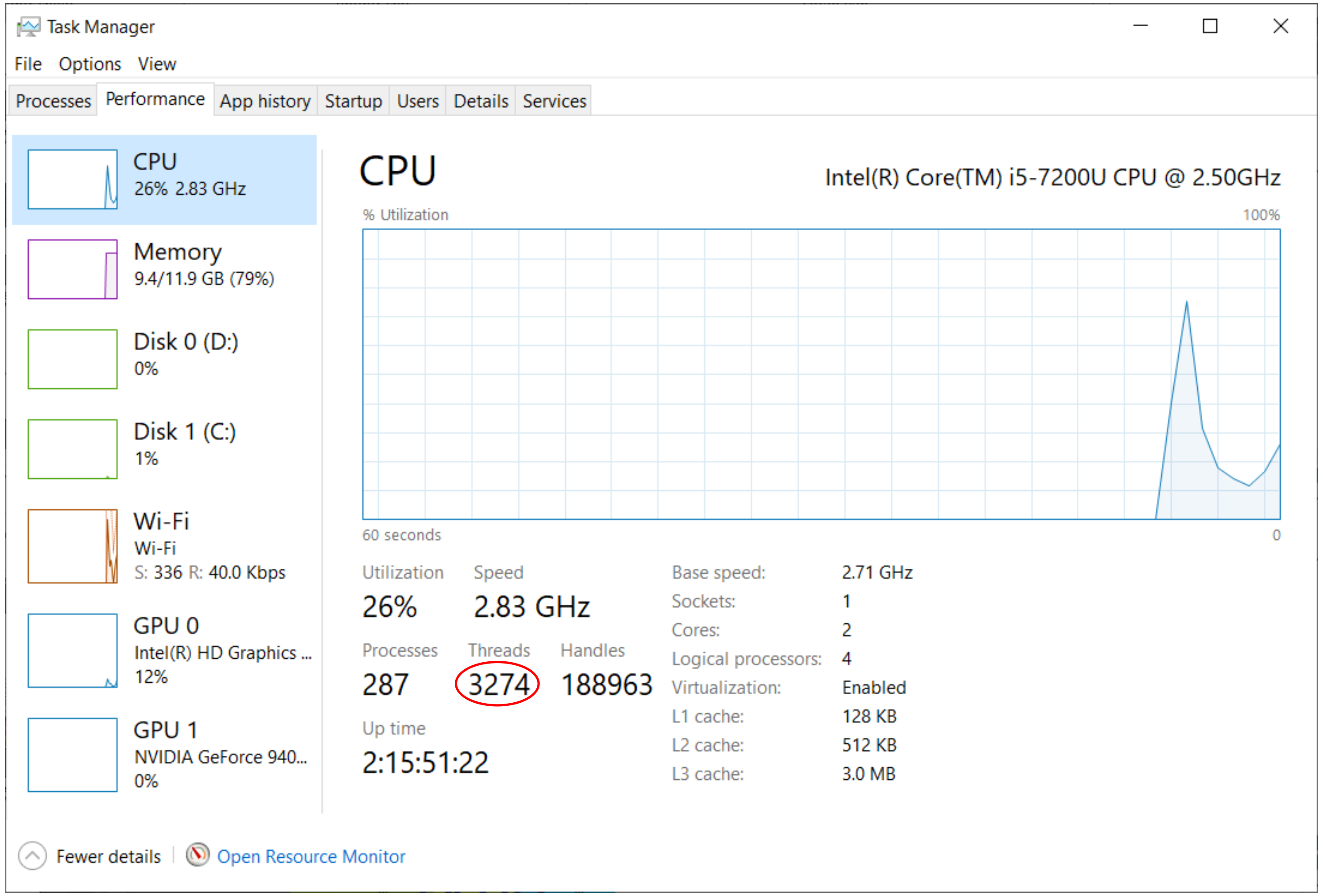


single-threaded process

multithreaded process



Thread



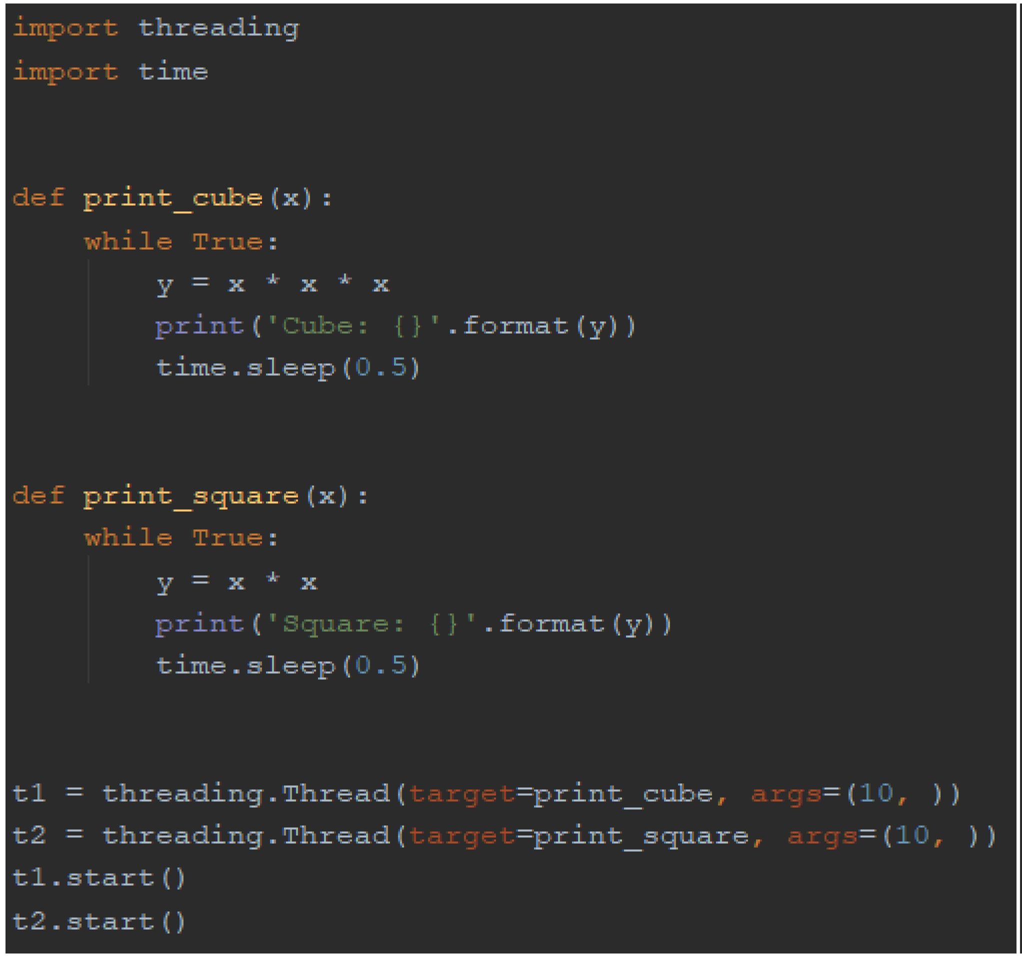
CPU 26% 2.83 GHz
Memory 9.4/11.9 GB (79%)
Disk 0 (D:) 0%
Disk 1 (C:) 1%
Wi-Fi Wi-Fi S: 336 R: 40.0 Kbps
GPU 0 Intel(R) HD Graphics ... 12%
GPU 1 NVIDIA GeForce 940... 0%

CPU Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz

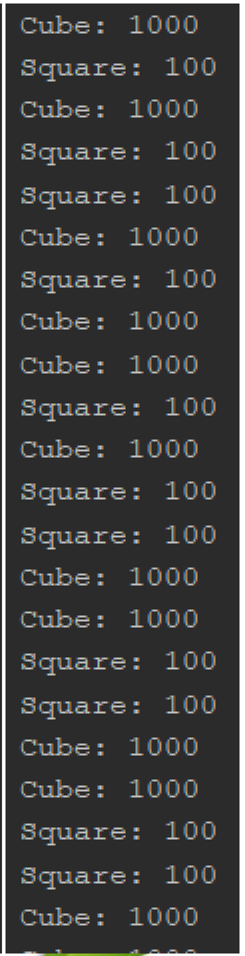
Utilization 26%
Speed 2.83 GHz
Processes 287
Threads 3274
Handles 188963
Up time 2:15:51:22

Base speed:	2.71 GHz
Sockets:	1
Cores:	2
Logical processors:	4
Virtualization:	Enabled
L1 cache:	128 KB
L2 cache:	512 KB
L3 cache:	3.0 MB

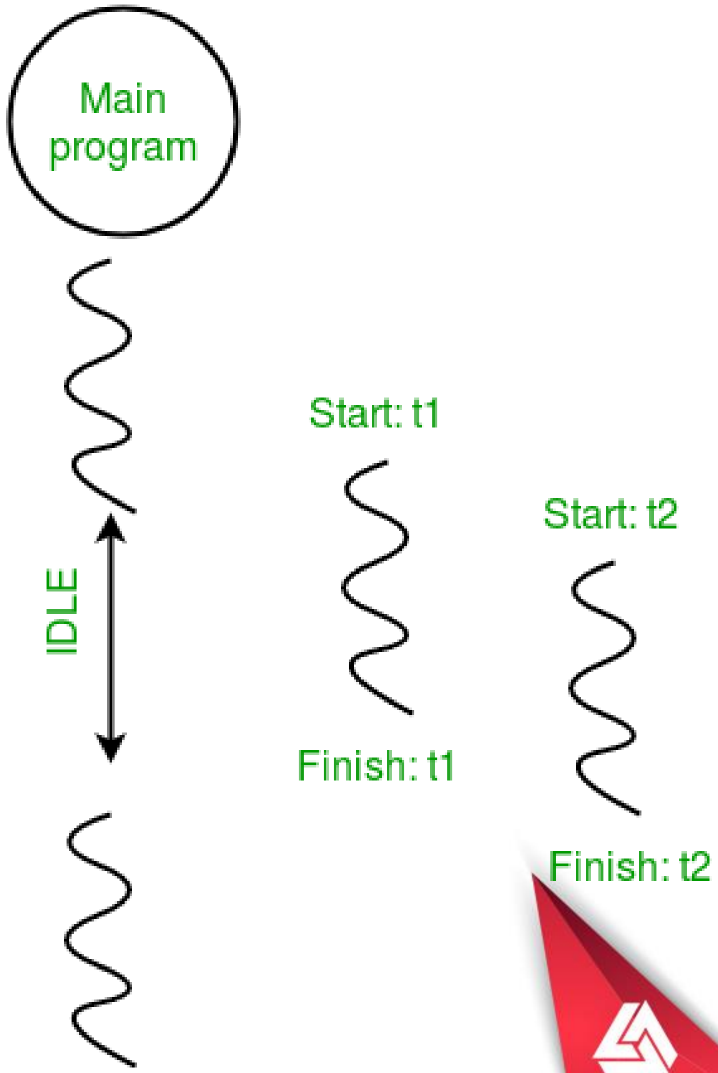
Thread



```
import threading
import time


def print_cube(x):
    while True:
        y = x * x * x
        print('Cube: {}'.format(y))
        time.sleep(0.5)


def print_square(x):
    while True:
        y = x * x
        print('Square: {}'.format(y))
        time.sleep(0.5)


t1 = threading.Thread(target=print_cube, args=(10, ))
t2 = threading.Thread(target=print_square, args=(10, ))
t1.start()
t2.start()
```

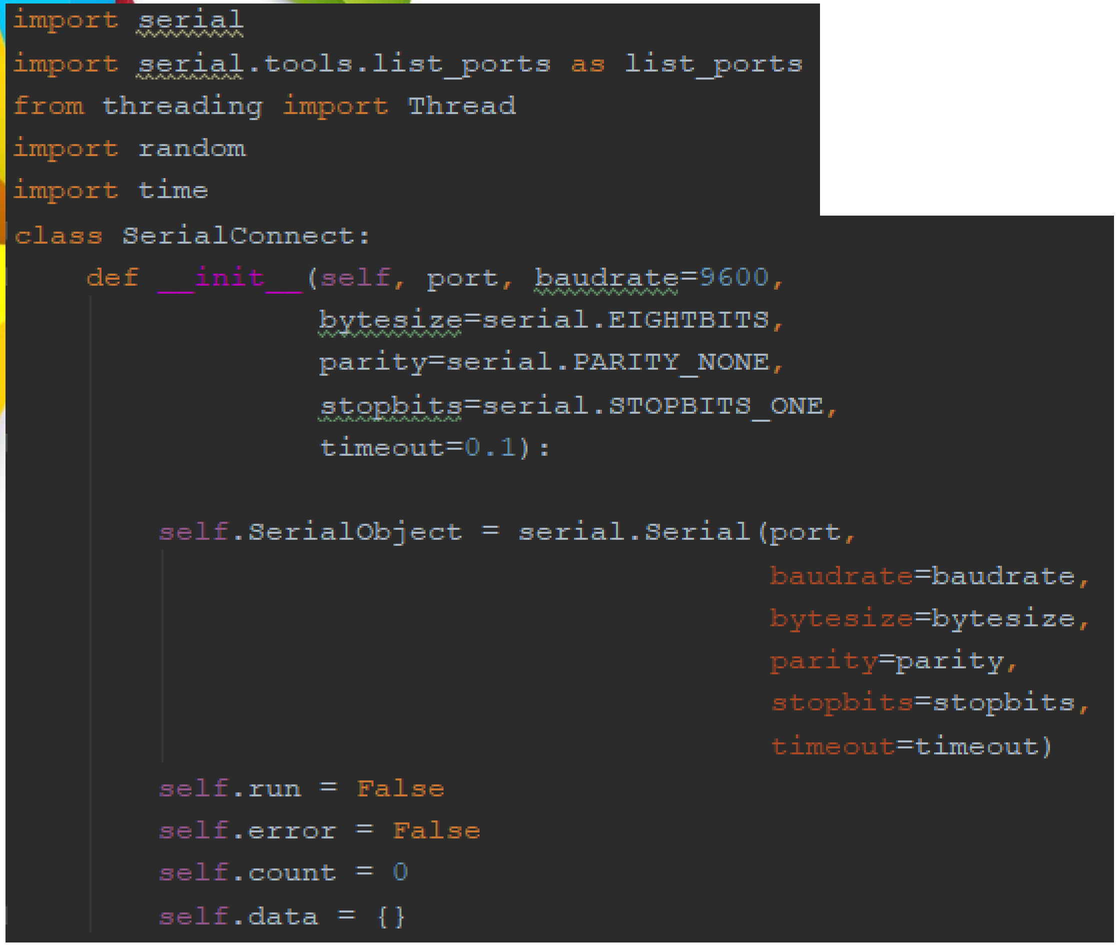


```
Cube: 1000
Square: 100
Cube: 1000
Square: 100
Square: 100
Cube: 1000
Square: 100
Cube: 1000
Cube: 1000
Square: 100
Cube: 1000
Square: 100
Square: 100
Cube: 1000
Cube: 1000
Square: 100
Square: 100
Cube: 1000
Cube: 1000
Square: 100
Square: 100
Cube: 1000
```

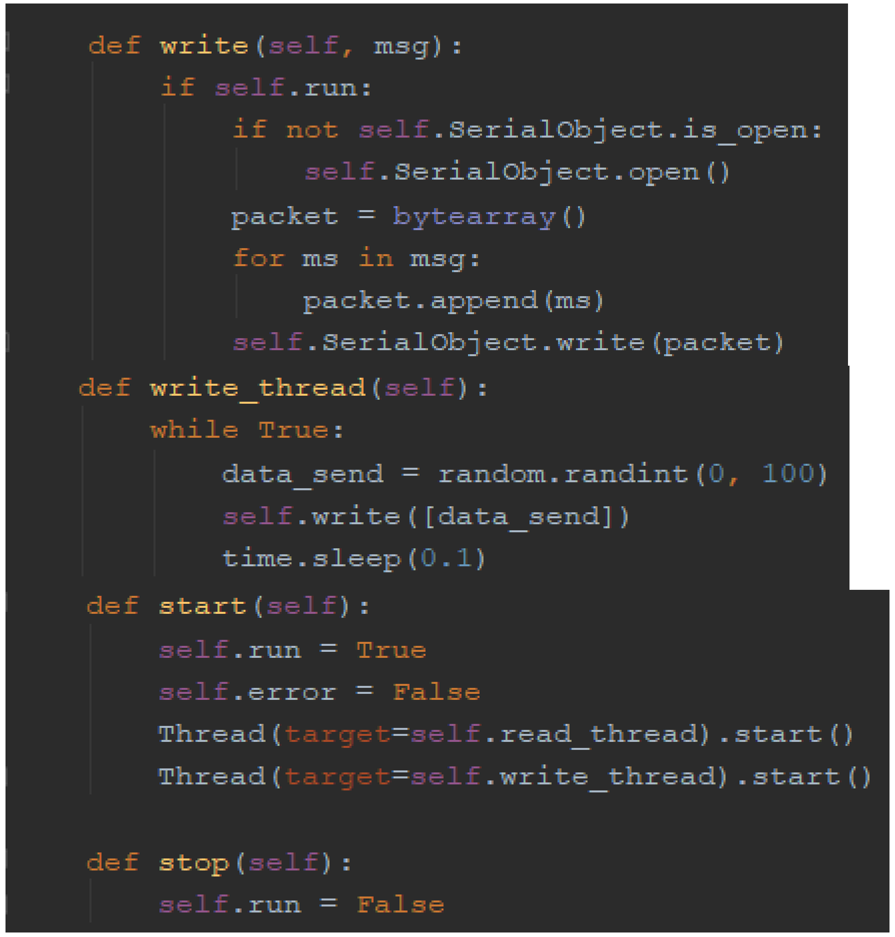


Pyserial

Tạo class cho toàn bộ tác vụ Serial áp dụng Thread (đa luồng)



```
import serial
import serial.tools.list_ports as list_ports
from threading import Thread
import random
import time
class SerialConnect:
    def __init__(self, port, baudrate=9600,
                 bytesize=serial.EIGHTBITS,
                 parity=serial.PARITY_NONE,
                 stopbits=serial.STOPBITS_ONE,
                 timeout=0.1):

        self.SerialObject = serial.Serial(port,
                                          baudrate=baudrate,
                                          bytesize=bytesize,
                                          parity=parity,
                                          stopbits=stopbits,
                                          timeout=timeout)
        self.run = False
        self.error = False
        self.count = 0
        self.data = {}
```

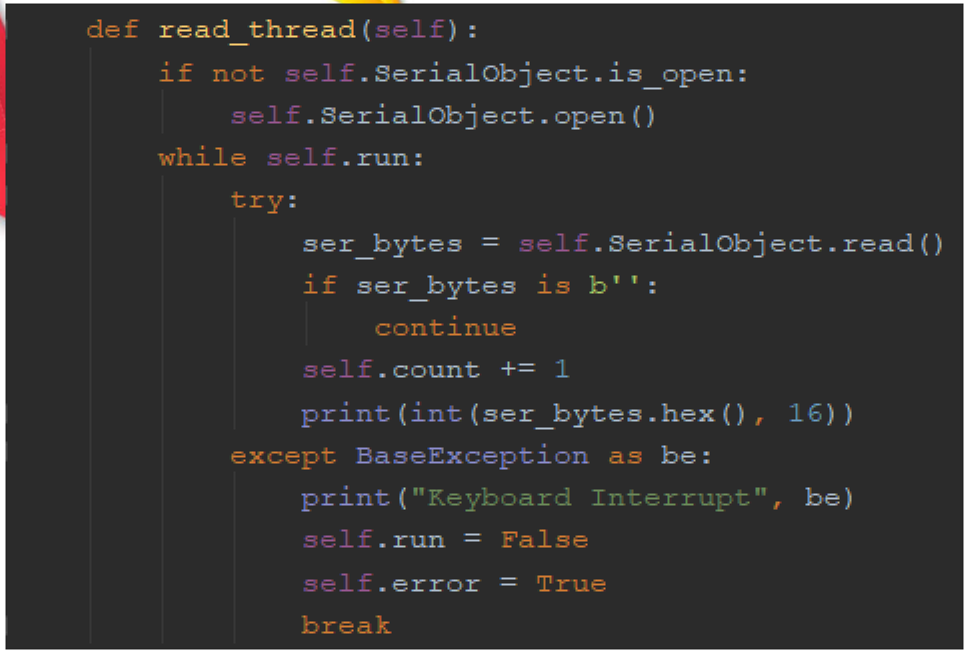


```
    def write(self, msg):
        if self.run:
            if not self.SerialObject.is_open:
                self.SerialObject.open()
            packet = bytearray()
            for ms in msg:
                packet.append(ms)
            self.SerialObject.write(packet)
    def write_thread(self):
        while True:
            data_send = random.randint(0, 100)
            self.write([data_send])
            time.sleep(0.1)
    def start(self):
        self.run = True
        self.error = False
        Thread(target=self.read_thread).start()
        Thread(target=self.write_thread).start()

    def stop(self):
        self.run = False
```

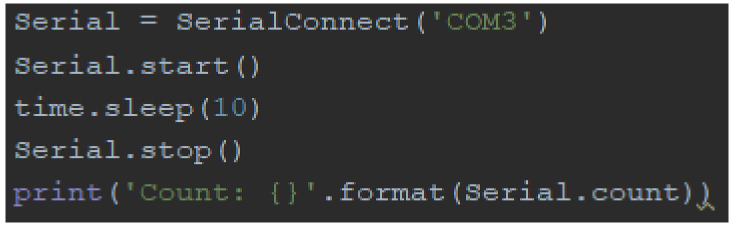
TAPIT.RGB

Pyserial

Tạo class cho toàn bộ tác vụ Serial áp dụng Thread (đa luồng)



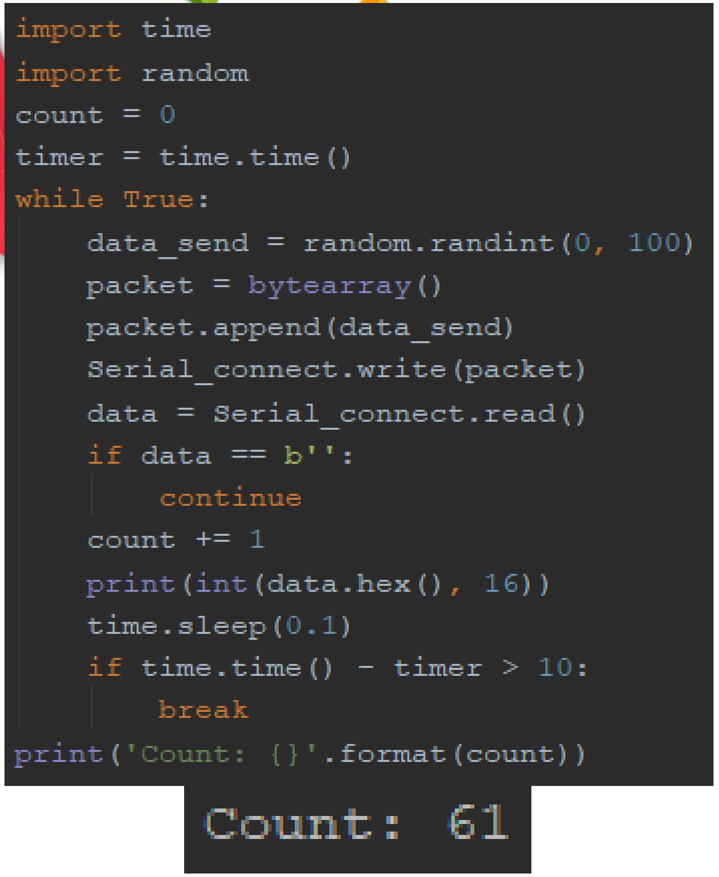
```
    def read_thread(self):
        if not self.SerialObject.is_open:
            self.SerialObject.open()
        while self.run:
            try:
                ser_bytes = self.SerialObject.read()
                if ser_bytes is b'':
                    continue
                self.count += 1
                print(int(ser_bytes.hex(), 16))
            except BaseException as be:
                print("Keyboard Interrupt", be)
                self.run = False
                self.error = True
                break
```



```
Serial = SerialConnect('COM3')
Serial.start()
time.sleep(10)
Serial.stop()
print('Count: {}'.format(Serial.count))
```

Pyserial

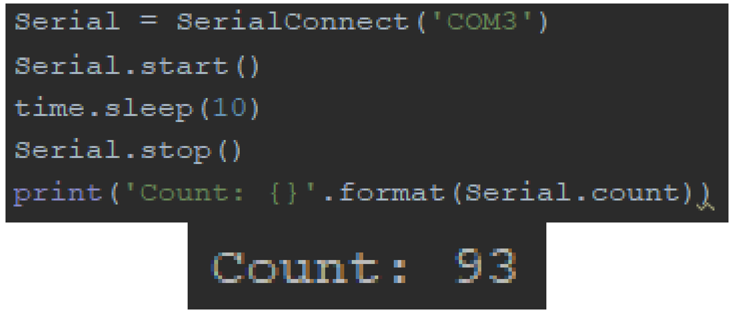
So sánh đơn luồng và đa luồng



```
import time
import random
count = 0
timer = time.time()
while True:
    data_send = random.randint(0, 100)
    packet = bytearray()
    packet.append(data_send)
    Serial_connect.write(packet)
    data = Serial_connect.read()
    if data == b'':
        continue
    count += 1
    print(int(data.hex(), 16))
    time.sleep(0.1)
    if time.time() - timer > 10:
        break
print('Count: {}'.format(count))
```

```
Count: 61
```

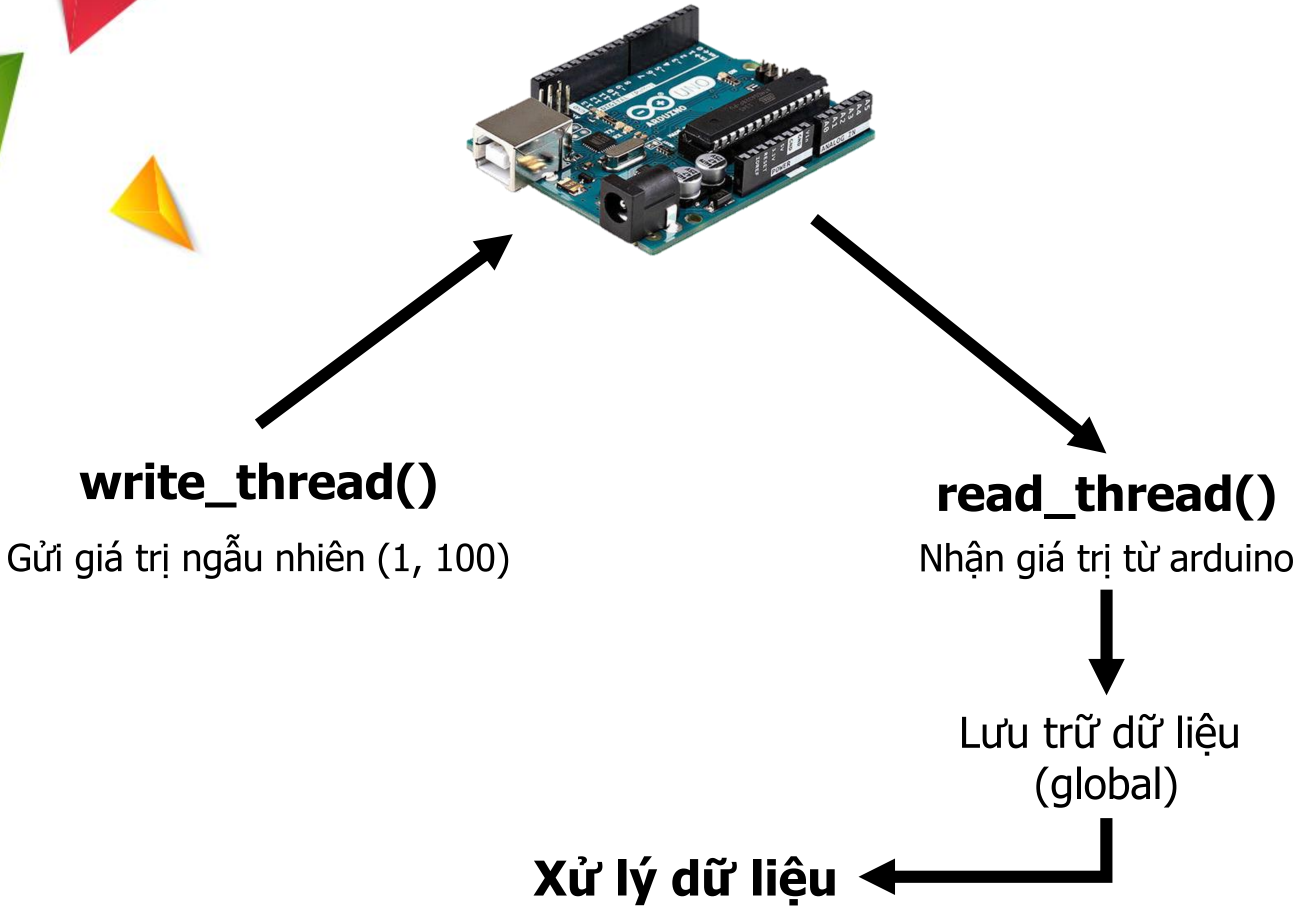
$<$



```
Serial = SerialConnect('COM3')
Serial.start()
time.sleep(10)
Serial.stop()
print('Count: {}'.format(Serial.count))
```

```
Count: 93
```

Queue

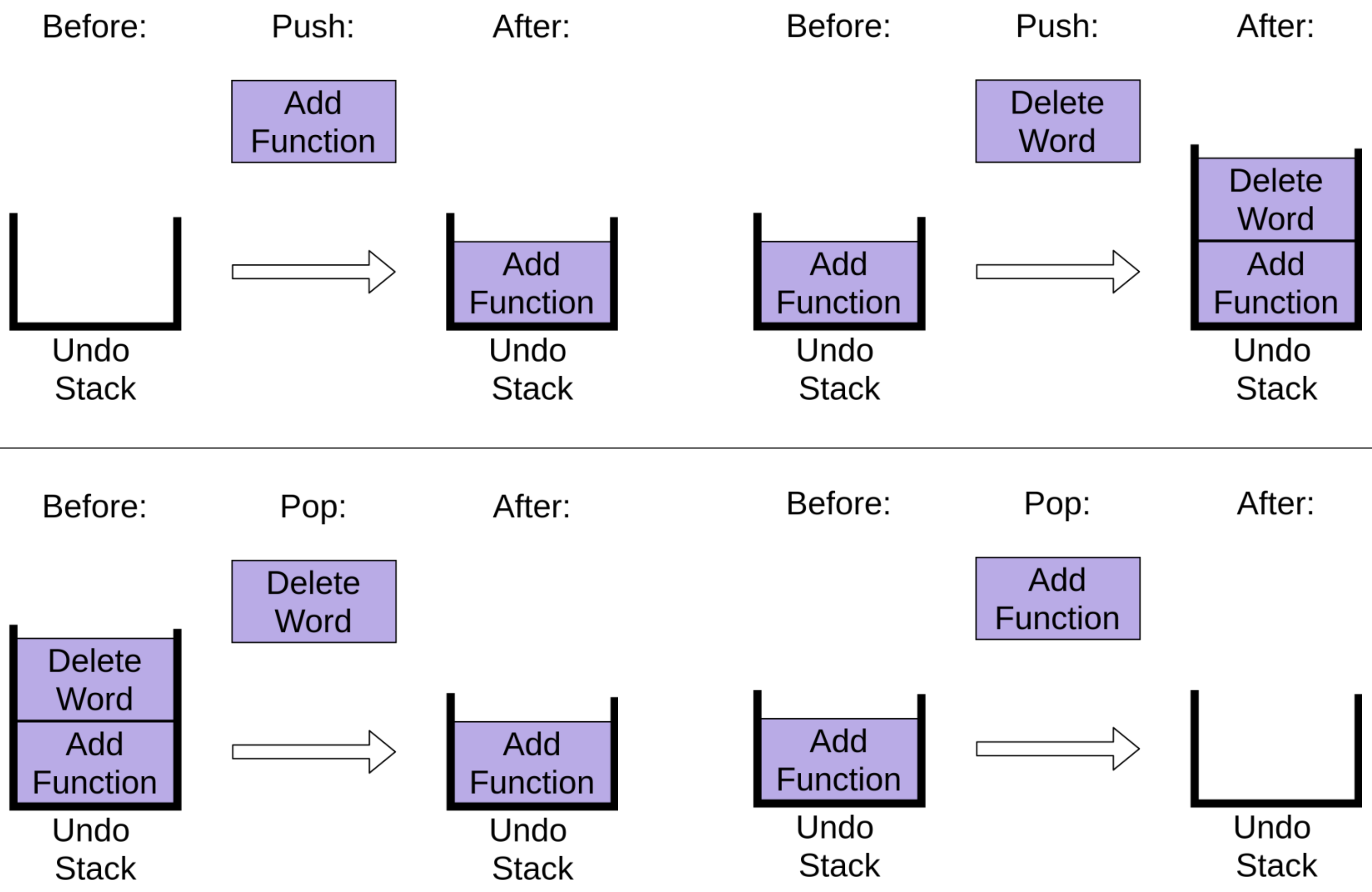

write_thread()
Gửi giá trị ngẫu nhiên (1, 100)
read_thread()
Nhận giá trị từ arduino
Lưu trữ dữ liệu (global)
Xử lý dữ liệu

Queue

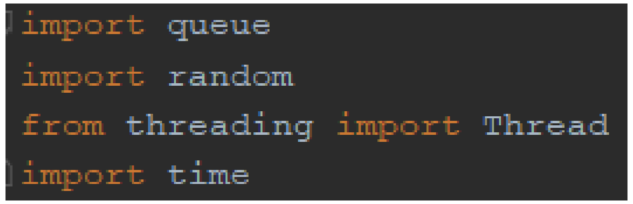
read_thread()

Nhận giá trị từ arduino

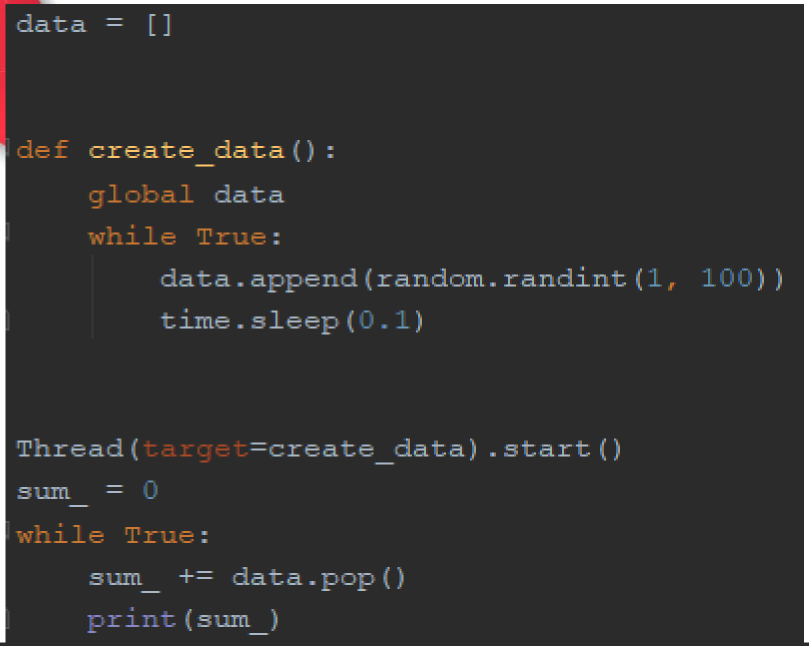
Xử lý dữ liệu



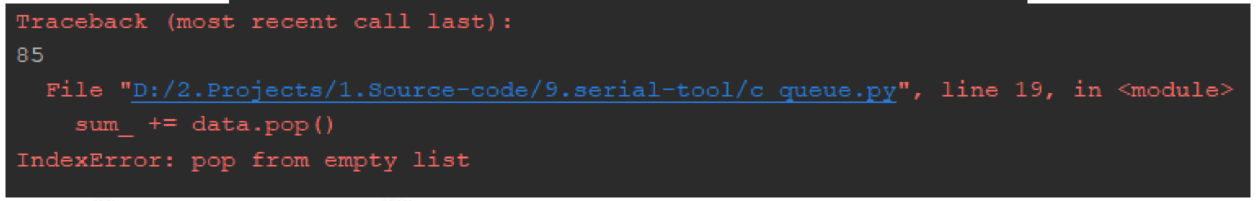
Queue



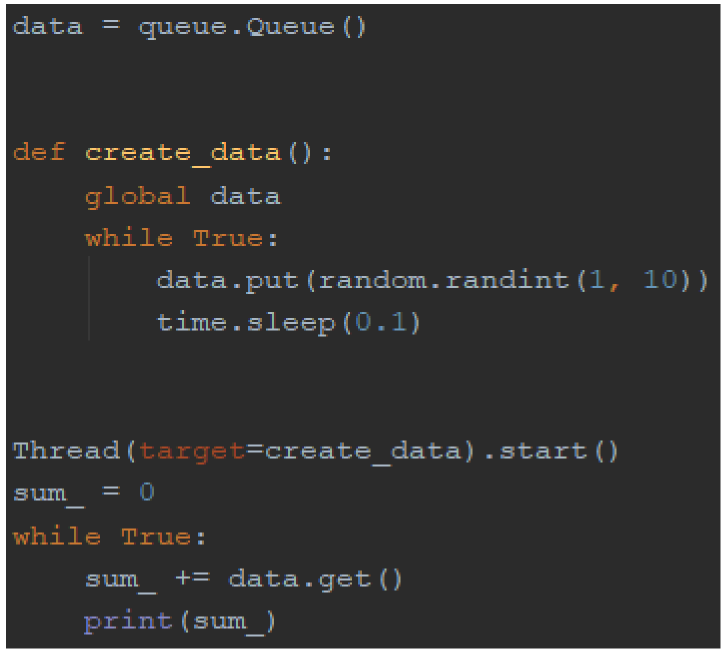
```python
import queue
import random
from threading import Thread
import time
```



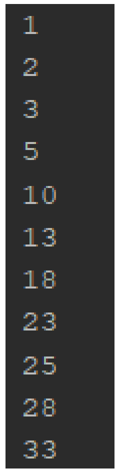
```python
data = []


def create_data():
    global data
    while True:
        data.append(random.randint(1, 100))
        time.sleep(0.1)


Thread(target=create_data).start()
sum_ = 0
while True:
    sum_ += data.pop()
    print(sum_)
```



```
Traceback (most recent call last):
85
  File "D:/2.Projects/1.Source-code/9.serial-tool/c queue.py", line 19, in <module>
    sum_ += data.pop()
IndexError: pop from empty list
```

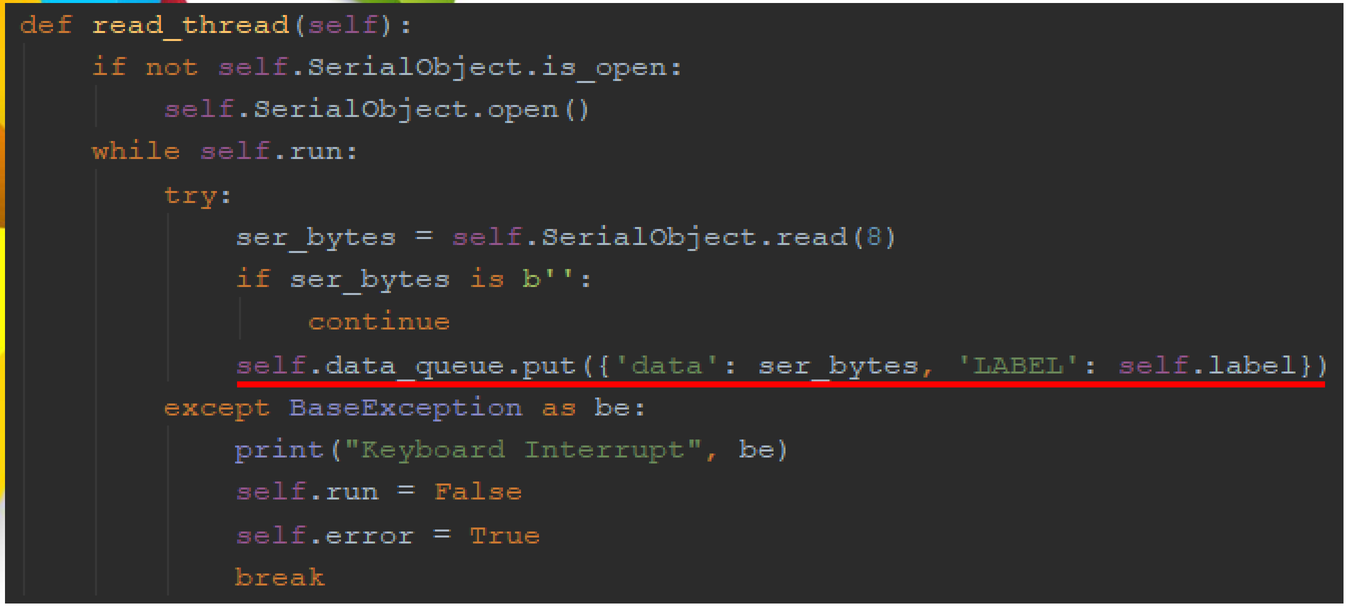


```python
data = queue.Queue()


def create_data():
    global data
    while True:
        data.put(random.randint(1, 10))
        time.sleep(0.1)


Thread(target=create_data).start()
sum_ = 0
while True:
    sum_ += data.get()
    print(sum_)
```

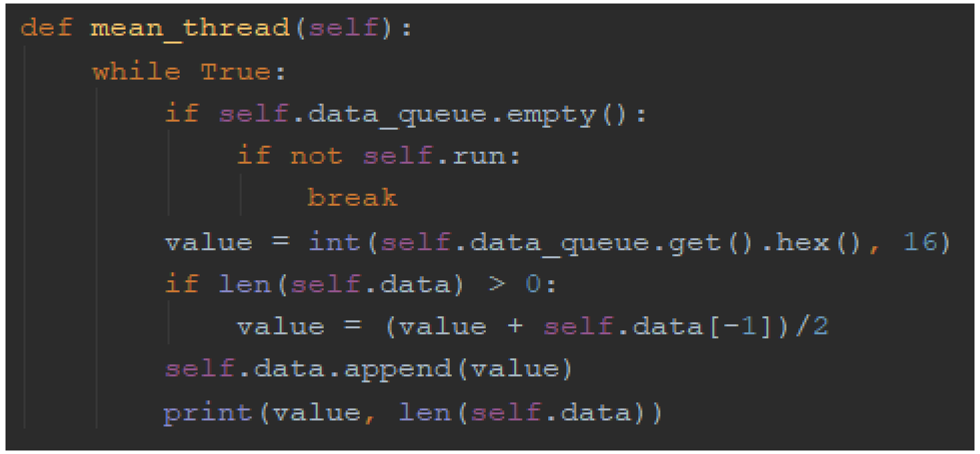


```
1
2
3
5
10
13
18
23
25
28
33
```

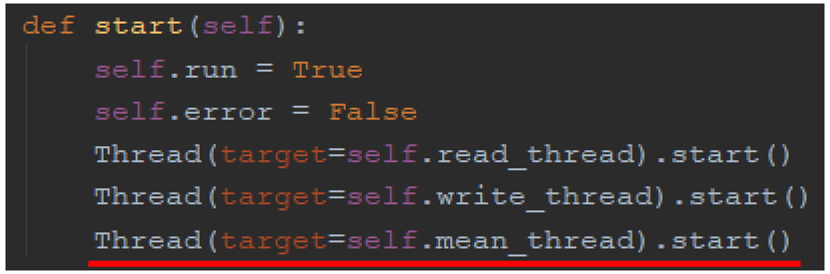
Ứng dụng Queue



```python
def read_thread(self):
    if not self.SerialObject.is_open:
        self.SerialObject.open()
    while self.run:
        try:
            ser_bytes = self.SerialObject.read(8)
            if ser_bytes is b'':
                continue
            self.data_queue.put({'data': ser_bytes, 'LABEL': self.label})
        except BaseException as be:
            print("Keyboard Interrupt", be)
            self.run = False
            self.error = True
            break
```

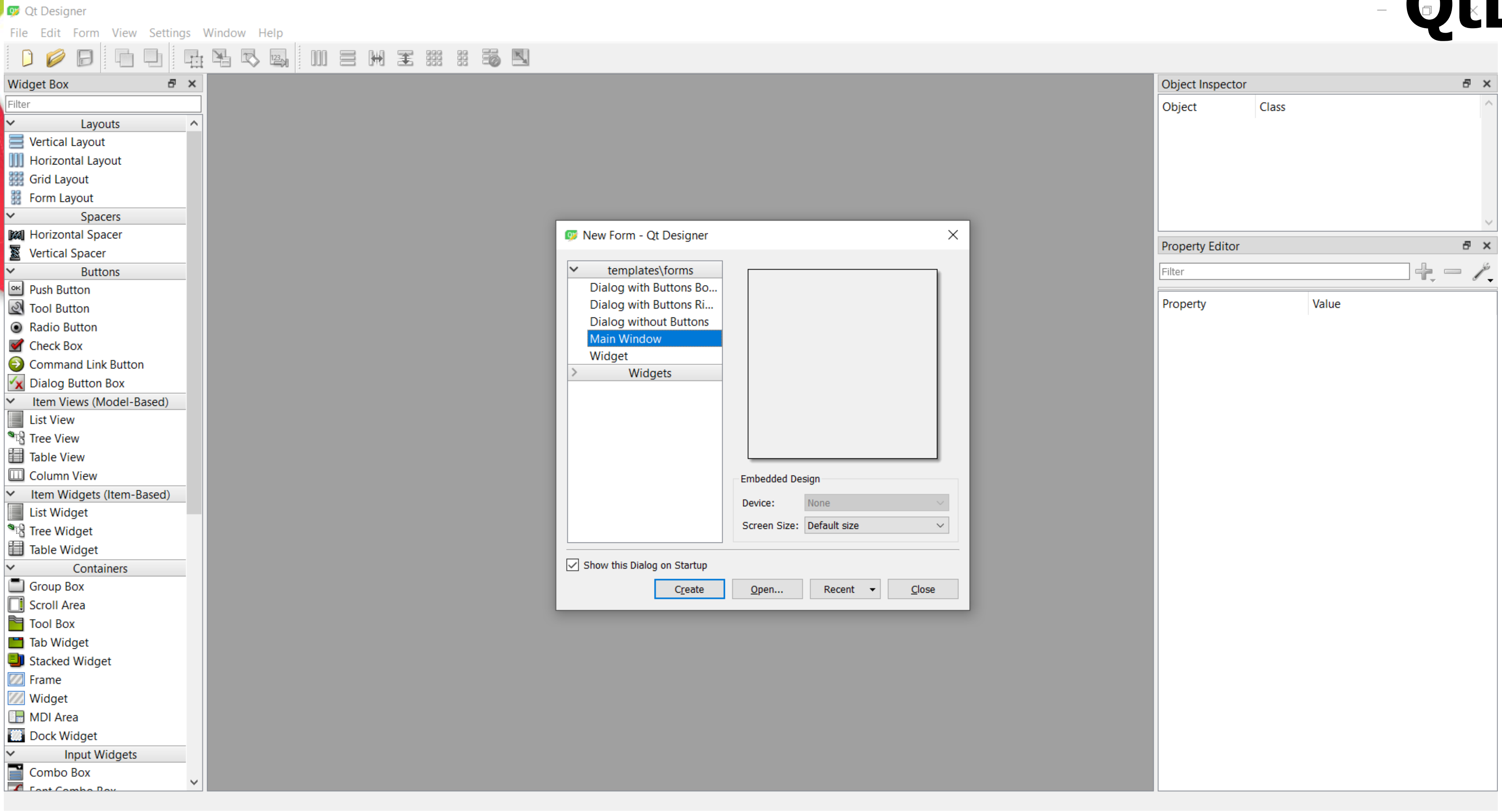


```python
def mean_thread(self):
    while True:
        if self.data_queue.empty():
            if not self.run:
                break
        value = int(self.data_queue.get().hex(), 16)
        if len(self.data) > 0:
            value = (value + self.data[-1])/2
        self.data.append(value)
        print(value, len(self.data))
```



```python
def start(self):
    self.run = True
    self.error = False
    Thread(target=self.read_thread).start()
    Thread(target=self.write_thread).start()
    Thread(target=self.mean_thread).start()
```

Pyqt QtDesign


Qt Designer

File Edit Form View Settings Window Help

Widget Box

Filter

Layouts
- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers
- Horizontal Spacer
- Vertical Spacer

Buttons
- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Based)
- List View
- Tree View
- Table View
- Column View

Item Widgets (Item-Based)
- List Widget
- Tree Widget
- Table Widget

Containers
- Group Box
- Scroll Area
- Tool Box
- Tab Widget
- Stacked Widget
- Frame
- Widget
- MDI Area
- Dock Widget

Input Widgets
- Combo Box

New Form - Qt Designer

templates\forms
- Dialog with Buttons Bo...
- Dialog with Buttons Ri...
- Dialog without Buttons
- Main Window
- Widget

Widgets

Embedded Design

Device: None

Screen Size: Default size

[✓] Show this Dialog on Startup

Create | Open... | Recent | Close

Object Inspector

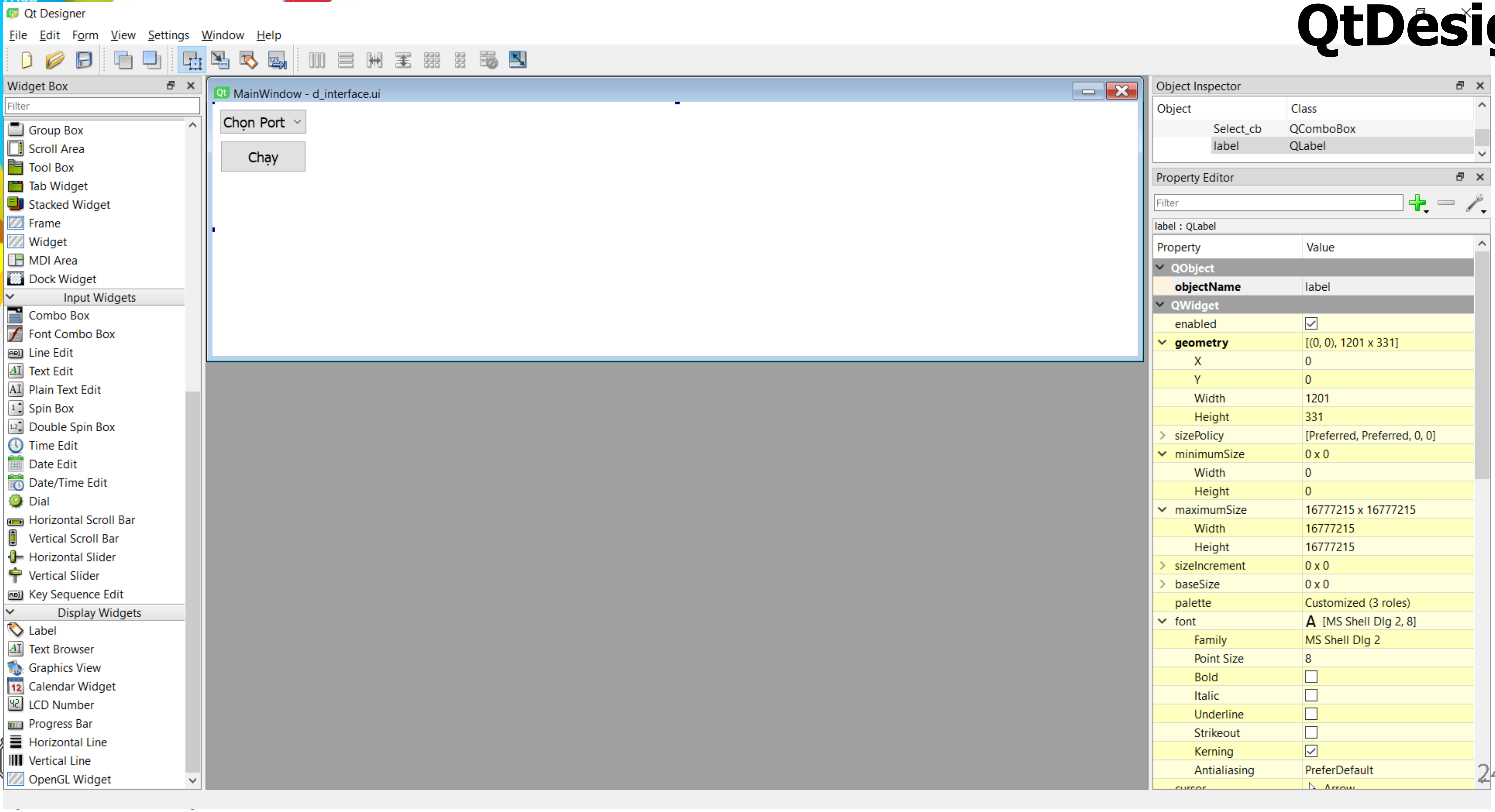
Object	Class

Property Editor

Filter

Property	Value

Pyqt QtDesign

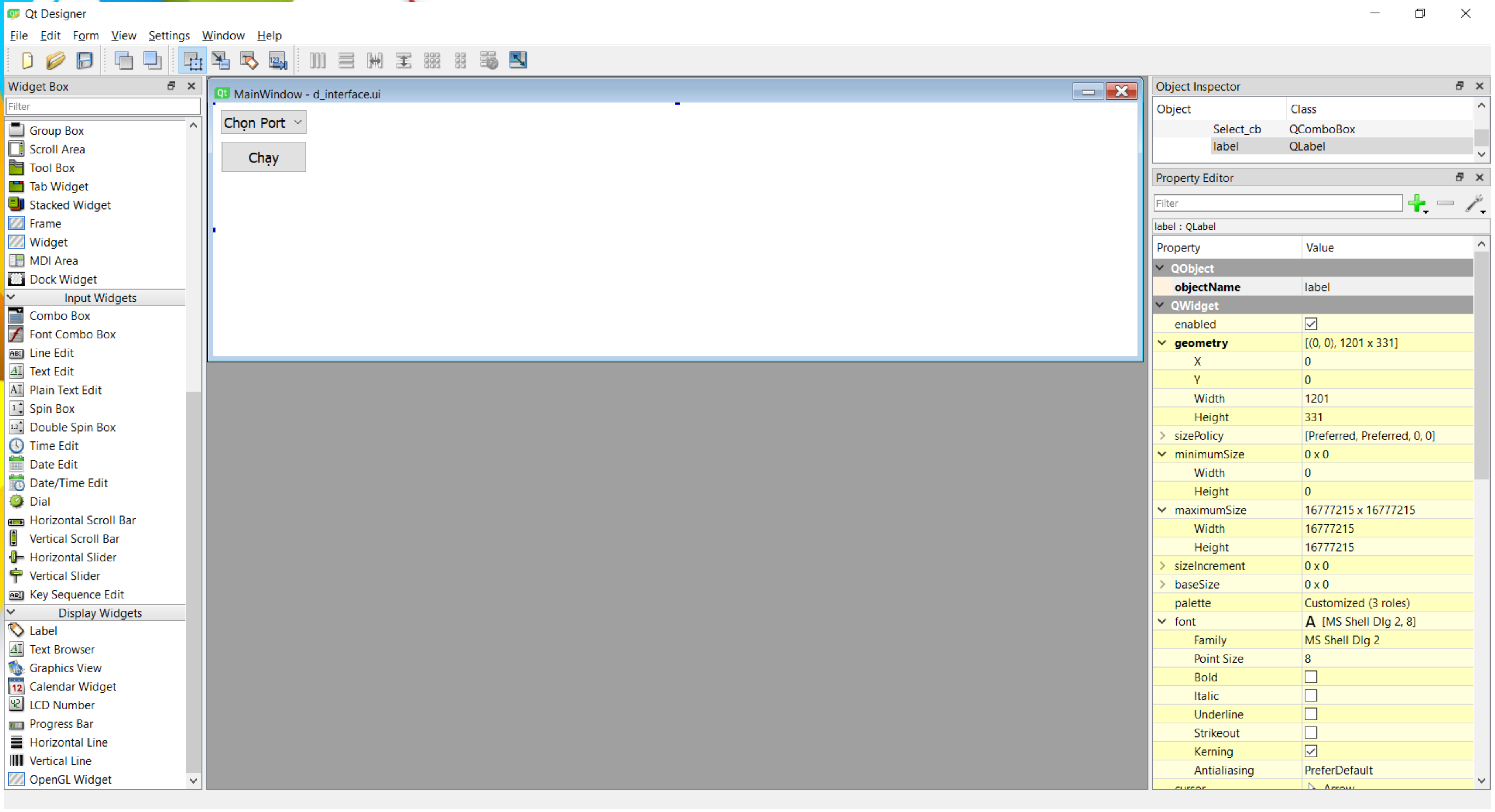


Object	Class
Select_cb	QComboBox
label	QLabel

label : QLabel

Property	Value
QObject	
objectName	label
QWidget	
enabled	☑
geometry	[(0, 0), 1201 x 331]
X	0
Y	0
Width	1201
Height	331
sizePolicy	[Preferred, Preferred, 0, 0]
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 16777215
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
palette	Customized (3 roles)
font	A [MS Shell Dlg 2, 8]
Family	MS Shell Dlg 2
Point Size	8
Bold	☐
Italic	☐
Underline	☐
Strikeout	☐
Kerning	☑
Antialiasing	PreferDefault

Pyqt QtDesign



Object	Class
Select_cb	QComboBox
label	QLabel

label : QLabel

Property	Value
QObject	
objectName	label
QWidget	
enabled	☑
geometry	[(0, 0), 1201 x 331]
X	0
Y	0
Width	1201
Height	331
sizePolicy	[Preferred, Preferred, 0, 0]
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 16777215
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
palette	Customized (3 roles)
font	A [MS Shell Dlg 2, 8]
Family	MS Shell Dlg 2
Point Size	8
Bold	☐
Italic	☐
Underline	☐
Strikeout	☐
Kerning	☑
Antialiasing	PreferDefault

```
>pyuic5.exe -x d_interface.ui -o d_interface.py
```