
Project Layout

my_project/

├── SerialManager.h

├── SerialManager.cpp

├── bindings.cpp

├── setup.py

└── test.py

File Contents

SerialManager.h

```
#pragma once
class SerialManager {
public:
    SerialManager();
    int hex_fn(unsigned char a, unsigned char b);
    int sum_fn(int x, int y);
    int multiply_fn(int x, int y);
};
```

SerialManager.cpp

```
#include "SerialManager.h"
SerialManager::SerialManager() {}
int SerialManager::hex_fn(unsigned char a, unsigned char b) {
    int temp = 0;
    temp = temp + b;
    temp = temp << 1;
    temp = temp + a;
    return temp;
}
int SerialManager::sum_fn(int x, int y) {
    return x + y;
```

```
}
int SerialManager::multiply_fn(int x, int y) {
    return x * y;
}
```

---

bindings.cpp

```
#include <pybind11/pybind11.h>
#include "SerialManager.h"
namespace py = pybind11;
PYBIND11_MODULE(my_module, m) {
    py::class_<SerialManager>(m, "SerialManager")
        .def(py::init<>())
        .def("hex_fn", &SerialManager::hex_fn)
        .def("sum_fn", &SerialManager::sum_fn)
        .def("multiply_fn", &SerialManager::multiply_fn);
}
```

---

setup.py

```
from setuptools import setup, Extension
import pybind11
ext_modules = [
    Extension(
        "my_module",
        ["bindings.cpp", "SerialManager.cpp"],
        include_dirs=[pybind11.get_include()],
        language="c++",
        extra_compile_args=["-std=c++17"],
    ),
]
setup(
    name="my_module",
```

```
    version="0.1",
    ext_modules=ext_modules,
)
```

---

test.py

```
import my_module
sm = my_module.SerialManager()
print("hex_fn:", sm.hex_fn(10, 20))
print("sum_fn:", sm.sum_fn(5, 7))
print("multiply_fn:", sm.multiply_fn(3, 4))
```

🛠 Build and Run

```
python3 setup.py build
python3 setup.py install --user
python3 test.py
```

Expected output:

```
hex_fn: 41
sum_fn: 12
multiply_fn: 12
```
