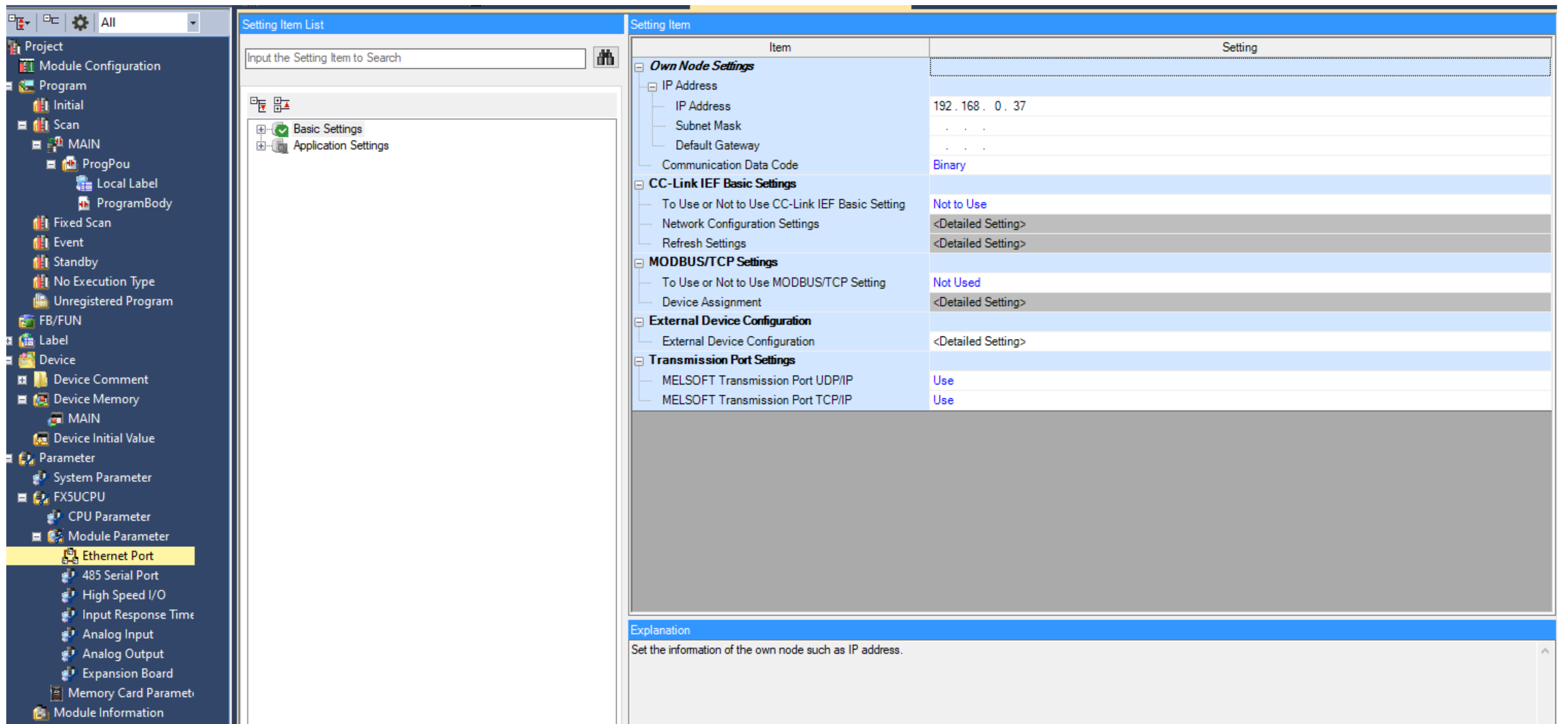


Hướng Dẫn Giao Tiếp PLC Mitsubishi FX5U VỚI Python



ETHERNET IP : 192.168.X.XX

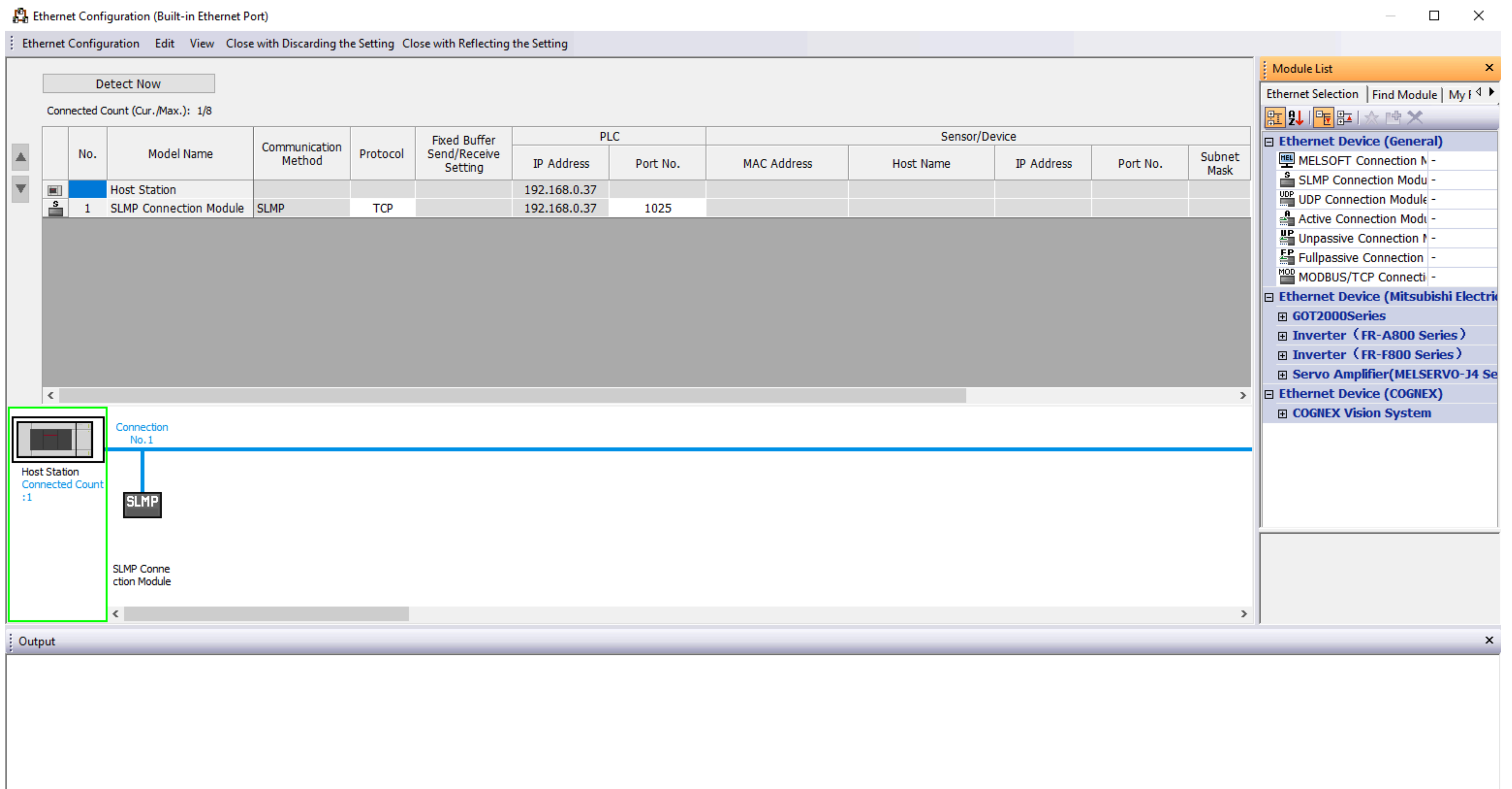


BƯỚC 1 : Tạo dự án mới với PLC FX5U

Project → Parameter → FX5UCPU → Modul Parameter → Sau khi vào đến đây chúng ta được như ảnh bên trên

BƯỚC 2 : Nhập địa chỉ IP cho PLC (Trùng 3 Lớp với IP của máy tính)

Nhập : 192.168.0.37 (Ví dụ) → Sau đó nháy double chuột vào External Decive Configurution



BƯỚC 4 : Tạo SLMP Connection Modul

Ethernet Decive (Góc Phải Trên Cùng) → Chọn SLMP Connection Modul (Dè chuột trái kéo ra) ---> Sau đó được như ảnh bên trên

BƯỚC 5 : Nhập địa chỉ Port No

Protocol : TCP

IP Address : 192.168.x.x (Giống Địa Chỉ PLC)

Port No : 1025

Sau đó được như ảnh bên trên !! Như vậy là xong phần setup với PLC giờ qua Pyhthon nha anh em !!



Hướng dẫn giao tiếp với PLC FX5U qua Ethernet

1. Cấu hình PLC

- Địa chỉ IP: 192.168.0.37
- Port: 1025 (TCP)
- Mã hóa dữ liệu: Binary

2. Cài đặt thư viện

Windows:

```
```bash
```

```
pip install rk_mcprotocol (py -m pip install rk_mcprotocol ,để cài nha anh em)
```

```
```
```

3. Các chức năng chính

3.1 Đọc dữ liệu

```
```python
```

```
Đọc bit từ M0-M3583 (giá trị 0 hoặc 1)
```

```
mc.read_bit(s, headdevice='m0', length=3584)
```

#### # Đọc word từ D0-D959

```
signed_type=True: giá trị từ -32,768 đến 32,767
```

```
signed_type=False: giá trị từ 0 đến 65,535
```

```
mc.read_sign_word(s, headdevice='d0', length=960, signed_type=True)
```





### **# Đọc double word từ R0-R479**

# signed\_type=True: giá trị từ -2,147,483,648 đến 2,147,483,647

# signed\_type=False: giá trị từ 0 đến 4,294,967,295

```
mc.read_sign_Dword(s, headdevice='r0', length=480, signed_type=True)
'''
```

### **### 3.2 Ghi dữ liệu**

```
```python
```

Ghi bit vào M0-M3583 (giá trị 0 hoặc 1)

```
mc.write_bit(s, headdevice='m0', data_list=[1]*3584)
```

Ghi word vào D0-D959

signed_type=True: giá trị từ -32,768 đến 32,767

signed_type=False: giá trị từ 0 đến 65,535

```
mc.write_sign_word(s, headdevice='d0', data_list=[100]*960,
signed_type=True)
```

Ghi double word vào R0-R479

signed_type=True: giá trị từ -2,147,483,648 đến 2,147,483,647

signed_type=False: giá trị từ 0 đến 4,294,967,295

```
mc.write_sign_Dword(s, headdevice='r0', data_list=[9999999]*480,
signed_type=True)
'''
```



4. Ví dụ sử dụng

```
```python
```

```
import rk_mcprotocol as mc
```

```
import time
```

```
def main():
```

```
 HOST = '192.168.0.37' # Địa chỉ IP của PLC
```

```
 PORT = 1025
```

```
 try:
```

```
 # Mở kết nối socket
```

```
 s = mc.open_socket(HOST, PORT)
```

```
 print(f"Đã kết nối thành công với PLC tại địa chỉ {HOST}")
```

```
 # Ghi bit M0-M7 ON
```

```
 print("\nGhi bit M0-M7 ON:")
```

```
 print(mc.write_bit(s, headdevice='m0', data_list=[1, 1, 1, 1, 1, 1, 1, 1]))
```

```
 # Đợi 1 giây
```

```
 time.sleep(1)
```

```
 # Ghi bit M0-M7 OFF
```

```
 print("\nGhi bit M0-M7 OFF:")
```

```
 print(mc.write_bit(s, headdevice='m0', data_list=[0, 0, 0, 0, 0, 0, 0, 0]))
```



## # Ghi giá trị 0 vào D0

```
print("\nGhi giá trị 0 vào D0:")
print(mc.write_sign_word(s, headdevice='d0', data_list=[0],
signed_type=False))

print("\nHoàn thành ghi dữ liệu!")

except Exception as e:
 print(f"Lỗi: {str(e)}")
finally:
 if 's' in locals():
 s.close()
 print("Đã đóng kết nối với PLC")

if __name__ == "__main__":
 main()
```



## ## 5. Giới hạn bộ nhớ mặc định của FX5U

| Loại thiết bị | Phạm vi mặc định | Hệ thống | Số điểm tối đa |

|-----|-----|-----|-----|

| X | X0 ~ X1777 | OCT | 1024 |

| Y | Y0 ~ Y1777 | OCT | 1024 |

| M | M0 ~ M7679 | DEC | 7680 |

| B | B0 ~ B0FF | HEX | 256 |

| L | L0 ~ L7679 | DEC | 7680 |

| F | F0 ~ F127 | DEC | 128 |

| D | D0 ~ D7999 | DEC | 8000 |

| W | W0 ~ W1FF | HEX | 512 |

| R | R0 ~ R32767 | DEC | 32768 |

## ## 6. Lưu ý quan trọng

1. Sử dụng mã hóa Binary thay vì ASCII để tăng tốc độ truyền thông
2. Không sử dụng Threading vì giao tiếp SLMP là half-duplex
3. Đảm bảo PLC đã được cấu hình đúng địa chỉ IP và port
4. Kiểm tra kết nối mạng trước khi thực hiện giao tiếp





# Chi tiết bài chia sẻ có trên Youtube



# YouTube

**FACTORY AUTOMATION**

**FACTORY AUTOMATION**