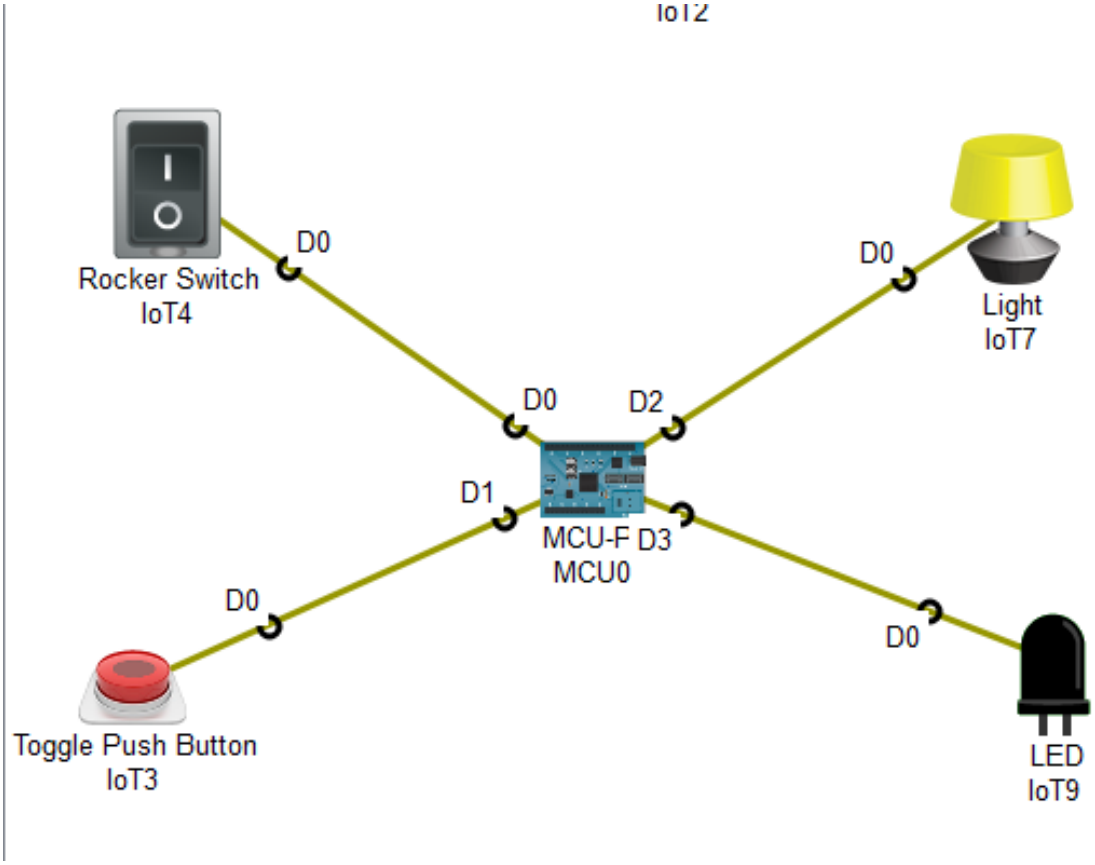
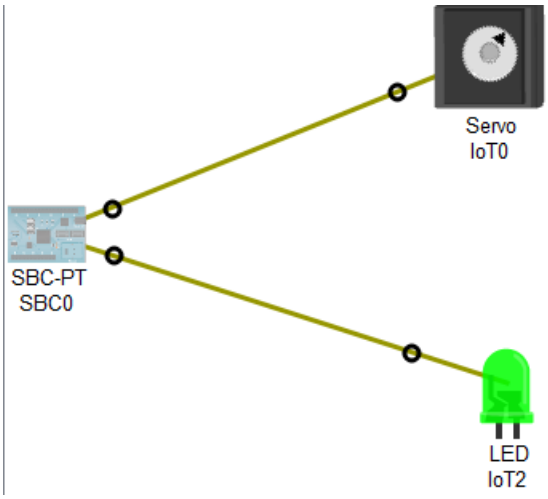
Microcontroller




```
from gpio import *
from time import *
switchValue = 0
togglePushButtonValue = 0
def readFromSensors():
        global switchValue
        global togglePushButtonValue
        switchValue = digitalRead (0)
        togglePushButtonValue = digitalRead(1)
def writeToActuators():
        if (switchValue == HIGH):
                digitalWrite(3, HIGH)
```

```
        else:
                digitalWrite(3, LOW)
        if (togglePushButtonValue == HIGH):
                customWrite (2, "2")
        else:
                customWrite(2, "0")
def main():
        pinMode (0, IN)
        pinMode (1, IN)
        pinMode (2, OUT)
        pinMode (3, OUT)
        pinMode (4, OUT)
        while True:
                readFromSensors()
                writeToActuators()
                delay(1000)
if __name__ == "__main__":
        main()
```

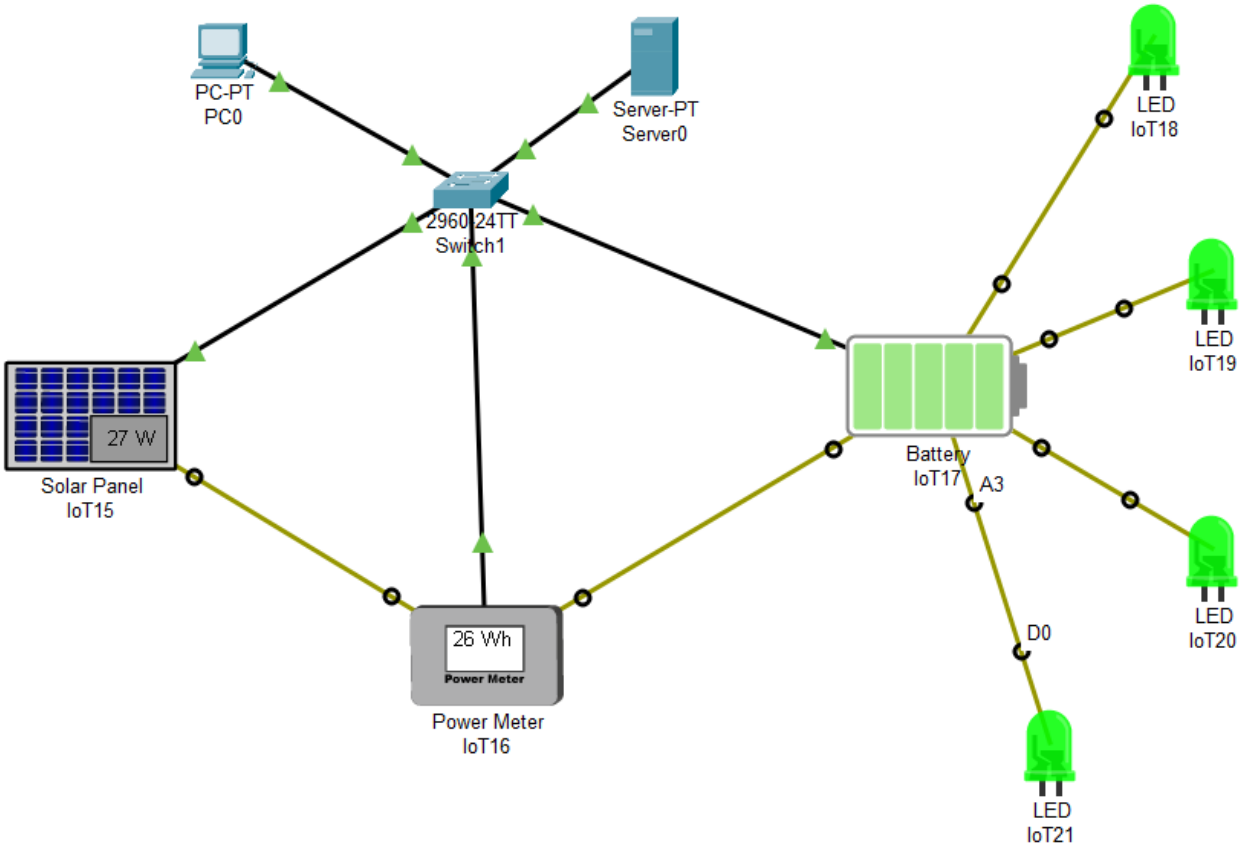
Servo




```
from gpio import *
from time import *

def main():
        pinMode(1, OUT)
        print("Blinking")
        while True:
                digitalWrite(1, HIGH);
                customWrite(0,127);
                delay(1000)
                digitalWrite(1, LOW);
                customWrite(0,-127);
                delay(500)

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

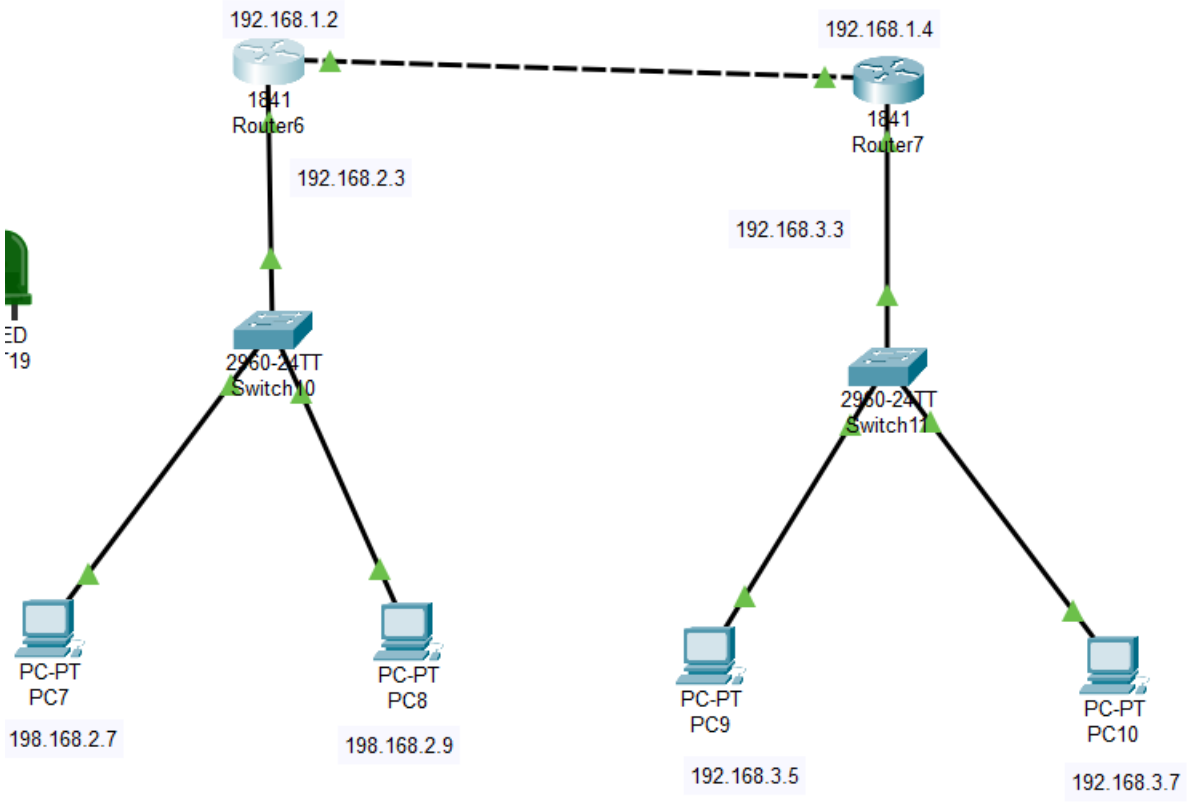


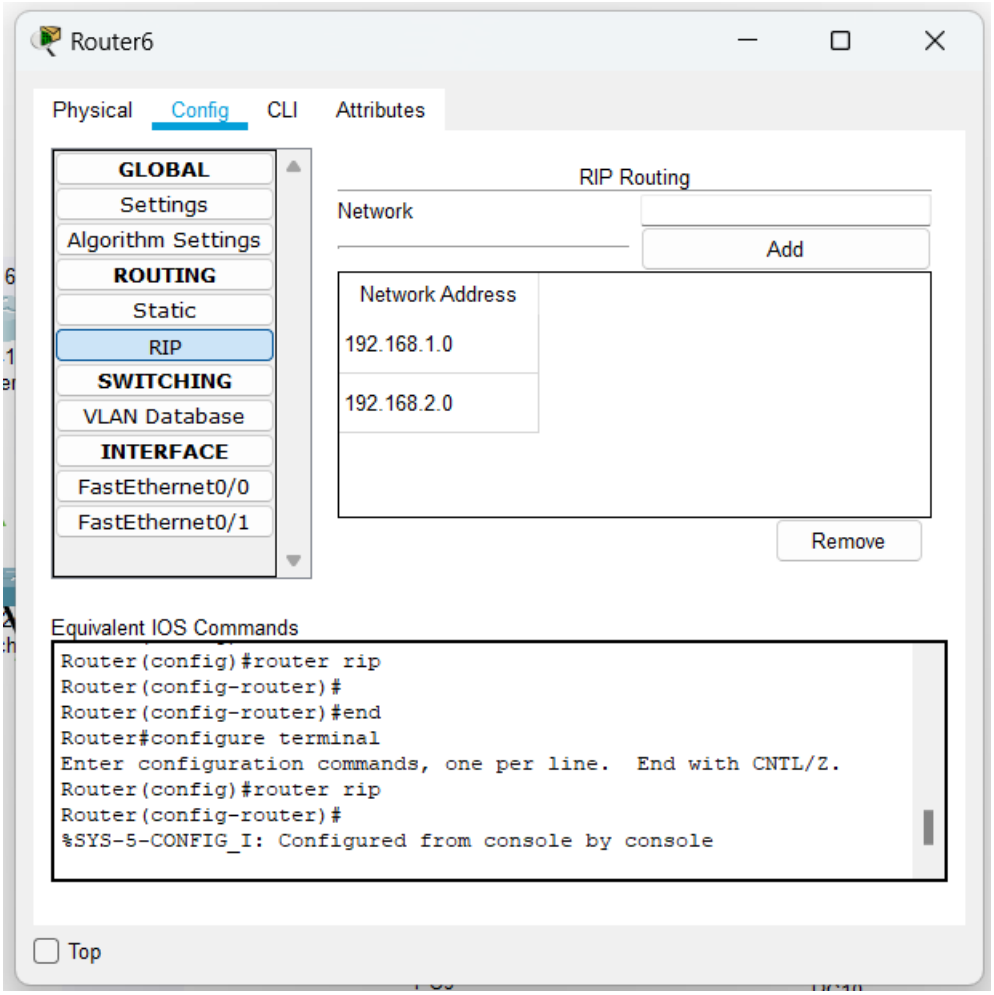
analogWrite(A0, availablePower);

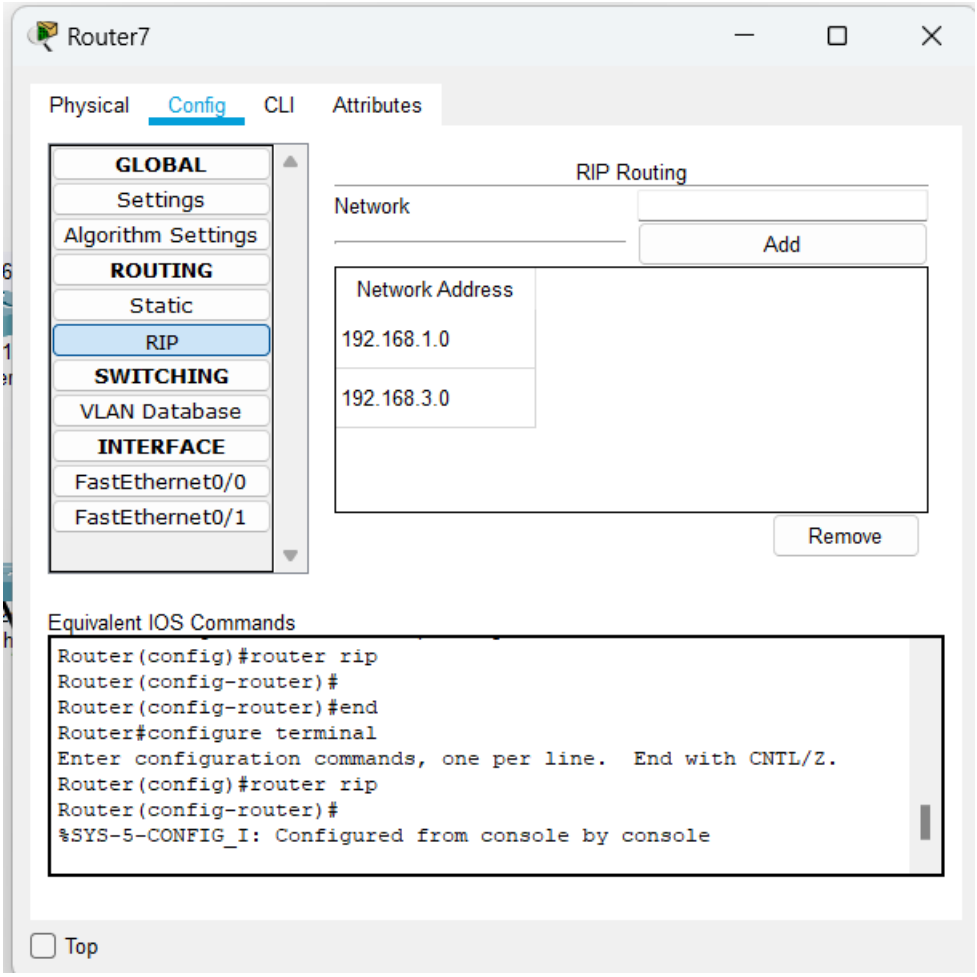
analogWrite(A1, availablePower);

analogWrite(A2, availablePower);

analogWrite(A3, availablePower)


192.168.1.2
192.168.1.4
1841
Router6
1841
Router7
192.168.2.3
192.168.3.3
2960-24TT
Switch10
2960-24TT
Switch11
PC-PT
PC7
198.168.2.7
PC-PT
PC8
198.168.2.9
PC-PT
PC9
192.168.3.5
PC-PT
PC10
192.168.3.7


Router6
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
RIP Routing
Network
Add
Network Address
192.168.1.0
192.168.2.0
Remove
Equivalent IOS Commands
Router(config)#router rip
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#
%SYS-5-CONFIG_I: Configured from console by console
Top


Router7
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
RIP Routing
Network
Add
Network Address
192.168.1.0
192.168.3.0
Remove
Equivalent IOS Commands
Router(config)#router rip
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#
%SYS-5-CONFIG_I: Configured from console by console
Top

TCP Client

```
import socket
sname="localhost"
port_number=8080
try:
    with socket.socket(socket.AF_INET,socket.SOCK_STREAM) as sock:
        sock.connect((sname,port_number))
        message="hi"
        sock.sendall(message.encode('utf-8'))
except Exception as e:
    print(e)
```

TCPServer

```
import socket
port_number=8080
with socket.socket(socket.AF_INET,socket.SOCK_STREAM) as sock:
    sock.bind(('localhost',port_number))
    sock.listen(1)
    client,adr = sock.accept()
    data=client.recv(1024).decode('utf-8')
    print(data)
    sock.close()
```

UDPClient

```
import socket
with socket.socket(socket.AF_INET,socket.SOCK_DGRAM) as sock:
    message="hi"
    sock.sendto(message.encode("utf-8"),('localhost',8080))
```

UDPServer

```
import socket
port_number=8080
with socket.socket(socket.AF_INET,socket.SOCK_DGRAM) as sock:
    sock.bind(('localhost',port_number))
    print("Listening....")
    data,adr = sock.recvfrom(1024)
    print(adr)
    print(data.decode('utf-8'))
    sock.close()
```