Marwadi University, Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Computer Networks (01CT0503)	**Aim: Design and simulate IoT scenario.**	
Experiment No: 13	**Date: 23-11-2025**	**Enrolment No: 92301733024**

Packet Tracer – Connecting Devices to Build IoT

Objectives
Get familiar with using Packet Tracer and its IoT connections.

Background / Scenario
In this activity you will build a connected solar-based power supply.
The sun charges the solar panel which sends electricity to the battery for power storage and distribution.
A power meter connected between them reads and display the amount of power being captured by the solar panel.
Because all devices are connected (IoT capabilities), they register themselves with a registration server, allowing a user to monitor the entire system from a web browser (running in the PC).

Required Resources
- Packet Tracer

Part 1: Adding and Connecting the Necessary Devices

You will start with four LEDs, a PC, a switch and a server. Add the following devices by locating and dragging them to Packet Tracer's work space:
Note: Light Emitting Diodes (LEDs) are electronic components used to emit light. LEDs are widely used in electronics as a form of user interaction.

a. A PT-Solar Panel device. PT-Solar Panel can be found under End Devices >> Power Grid.
b. A PT-Battery device. PT-Battery can be found under End Devices >> Power Grid.
c. A PT-Power Meter device. PT-Power Meter can be found under End Devices >> Power Grid.
d. Using IoT Custom Cables, connect the solar panel and the battery to the power meter according to the table below. The IoT Custom Cable can be found under Connections.
Use the table below to find the correct ports:

Device	Port	Power Meter Port
Solar Panel	D0	D0
Battery	D0	D1

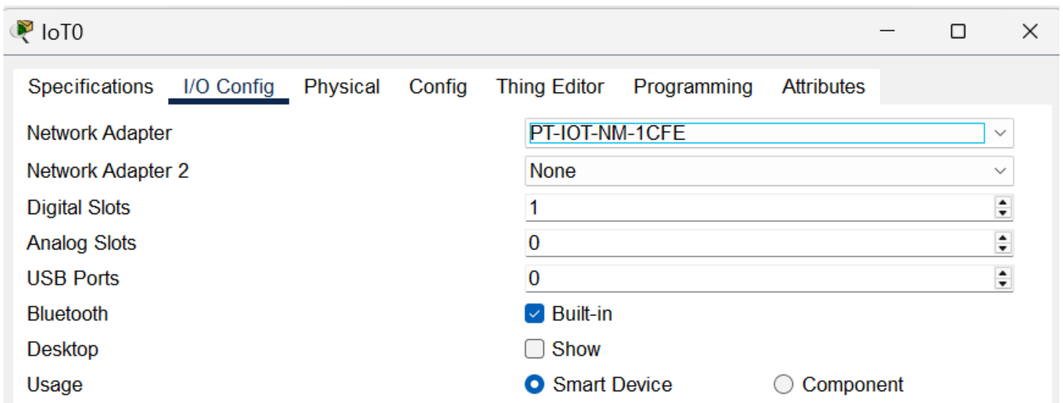
e. Using IoT Custom Cables, connect the LEDs to the battery according to the table below. The IoT Custom Cable can be found under Connections.
Use the table below to find the correct ports:

Device	Battery Port
LED1	D1
LED2	D2
LED3	D3
LED4	D4

Note: Packet Tracer may name IoT devices differently. While the name will not impact the activity, feel free to rename your devices for easy identification.
Note: For simplicity, Packet Tracer does not implement power cable properly. Wiring and powering devices involve concepts such as ground, polarity, specific connectors, cable width and more. For simplicity, Packet Tracer hides all these variables behind the multi-use cable IoT Custom Cable. In real

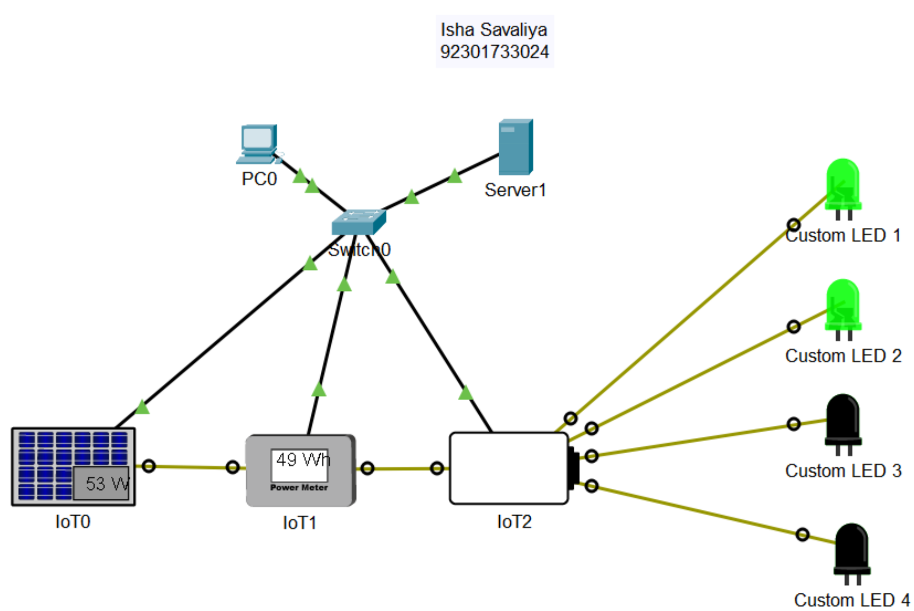
world designs, make sure to select the proper cable and connectors

f. Since in Packet Tracer IoT devices are wireless by default, to cable an IoT device to the network, you need to change the network adapter. Click the Solar Panel, and then click Advanced at the bottom right corner. More tabs are now available. Click the I/O Config tab. In the drop-down menu for Network Adapter, change it to PT-IOT-NM-1CFE for a FastEthernet connection. Repeat the process for the Power Meter and Battery.



g. Using an Ethernet straight-through cable, connect the Ethernet port of the solar power, battery, and power meter to the switch according the table below. This is to ensure they can communicate with the Server. Ethernet straight-through cables can be found under Connections.

Device	Switch Port
Solar Panel	Fa0/3
Power Meter	Fa0/4
Battery	Fa0/5



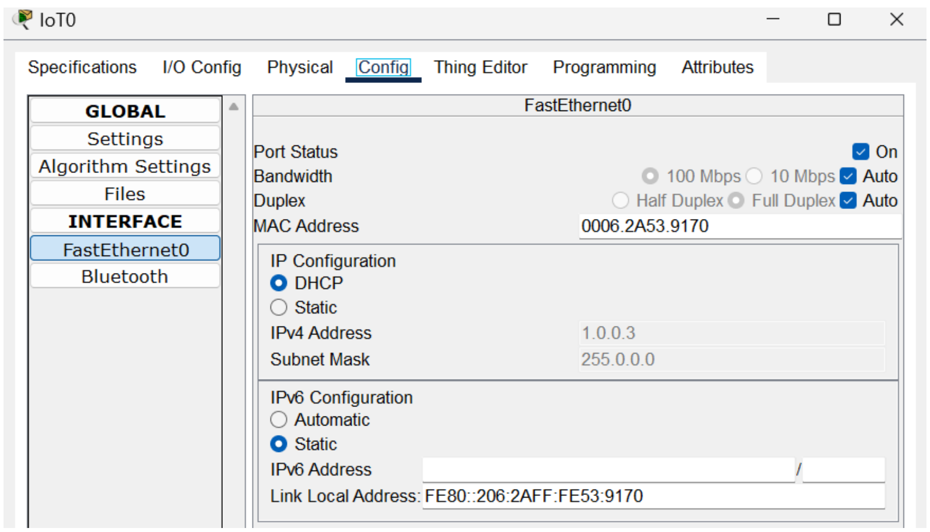
Part 2 : Configuring The Devices

Now that the devices are properly cabled, they must be configured. Since this system relies on an IP network, the devices must be configured with correct IP information. Since the server is configured to also act as a DHCP server, the IoT devices should be configured as DHCP clients in order to learn IP information automatically.

a. Click the solar panel, navigate to the Config tab >> GigabitEthernet0 and select DHCP under IP Configuration.

b. Click the power meter, navigate to the Config tab >> FastEthernet0 and select DHCP under IP Configuration.

c. Click the battery, navigate to the Config tab >> FastEthernet0 and select DHCP under IP Configuration.



What IP addresses were learned by the solar panel, power meter and battery?

- ➢ The IP addresses assigned were the DHCP addresses obtained from the server.

d. Before the device can operate properly, they must register to the server. Configure the devices with the server's IP address to allow them to find and communicate to the server.

Click the solar panel, navigate to the Config tab >> Settings and select Remote Server under IoT Server. Enter the following server information:

Server Address: 1.0.0.1

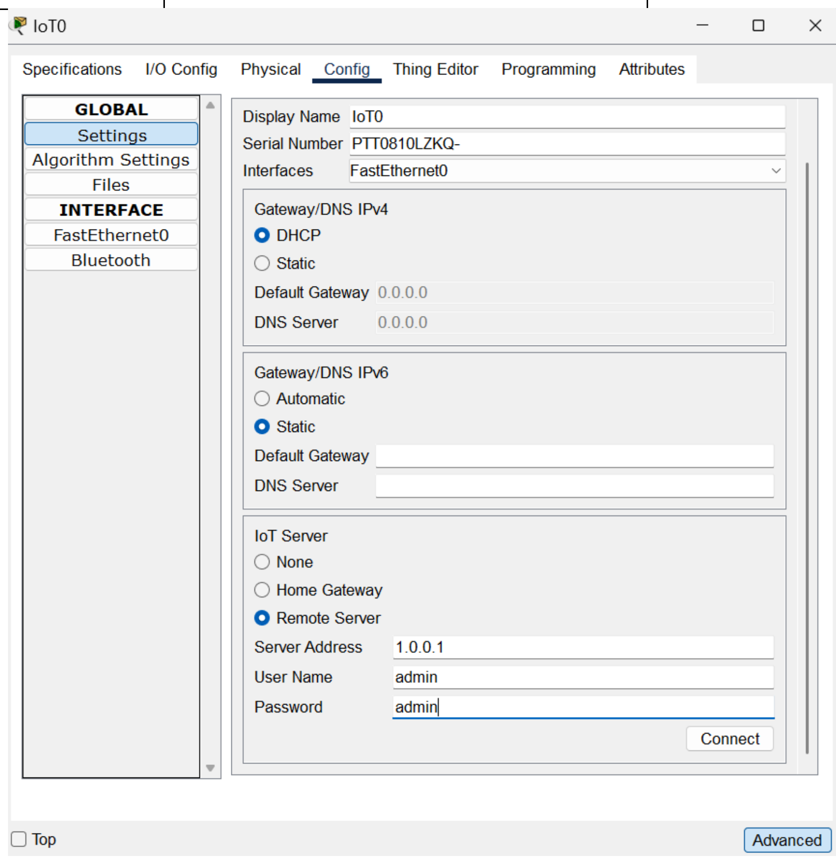
Username: admin

Password: admin

e. Click Connect.

f. Repeat the process for the power meter and battery. Use the same server address, username, and password as shown above.

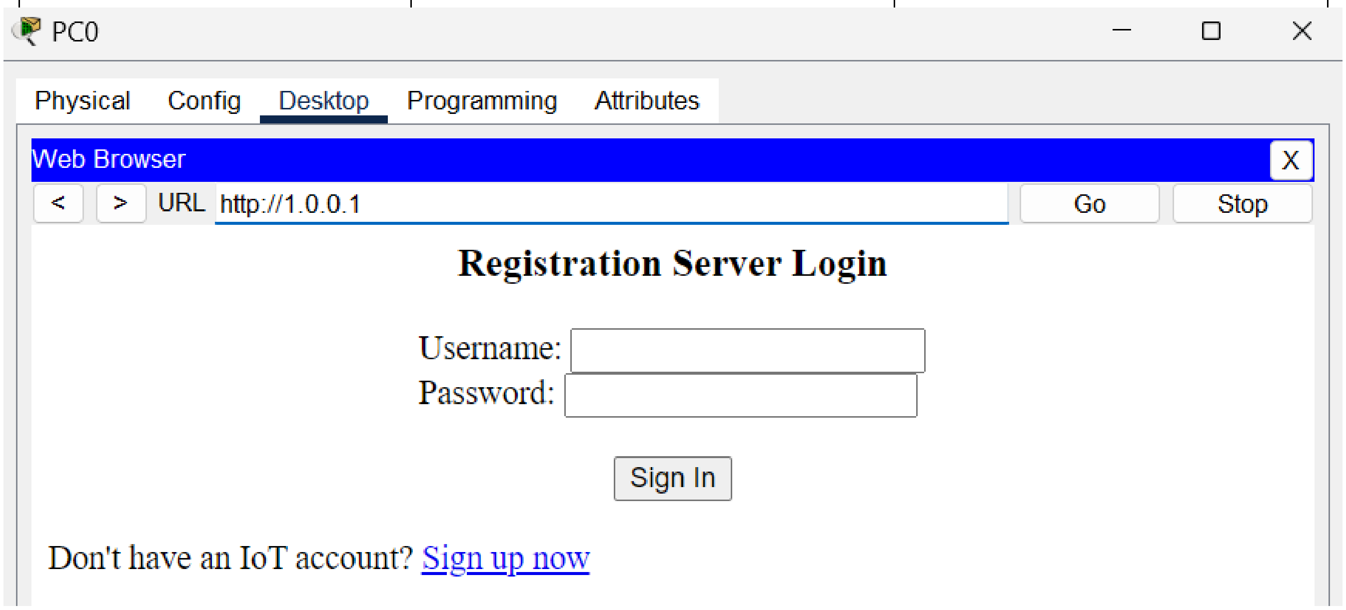
Marwadi University, Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Computer Networks (01CT0503)	**Aim: Design and simulate IoT scenario.**	
Experiment No: 13	**Date: 23-11-2025**	**Enrolment No: 92301733024**

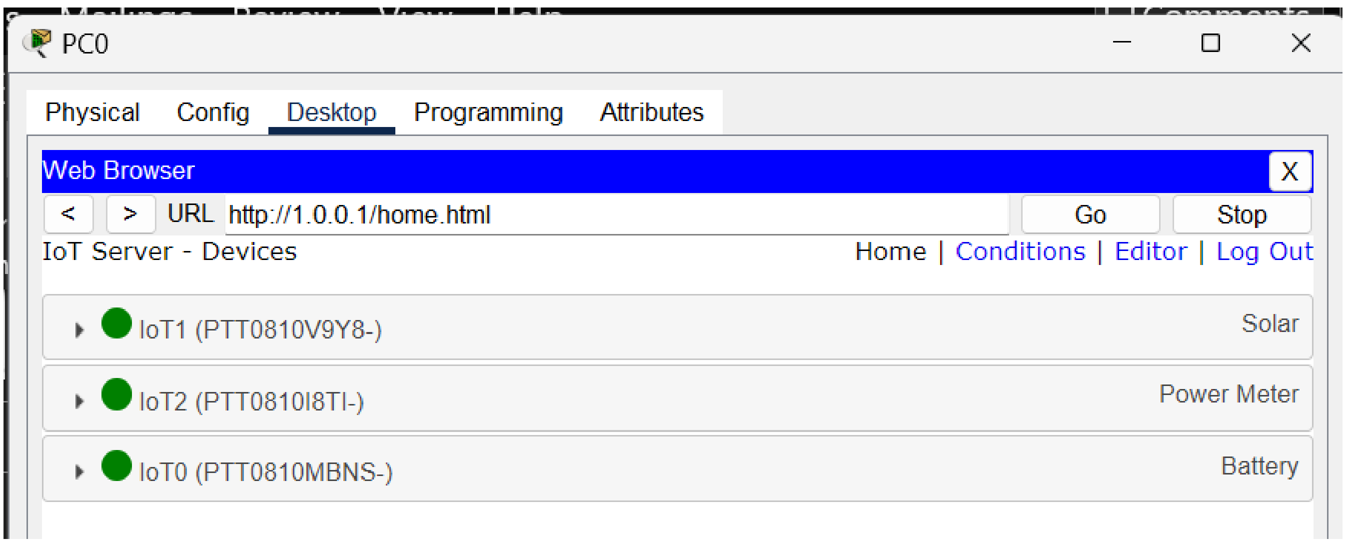


Part 3: Using the System

a. Now that all devices are connected, notice how the solar panel charges the battery.
b. Notice how the LEDs draw power from battery for operation.
c. Notice how the LEDs go dark if the battery has no charge.
d. Click the PC and navigate to Desktop >> Web Browser.
e. Type the IP address of the server, 1.0.0.1, and press enter.
f. Use the following credentials to log into the server:

Username: admin
Password: admin





How many devices are displayed in the page? What are their names?

- ➢ Solar Panel, Power Meter, Battery

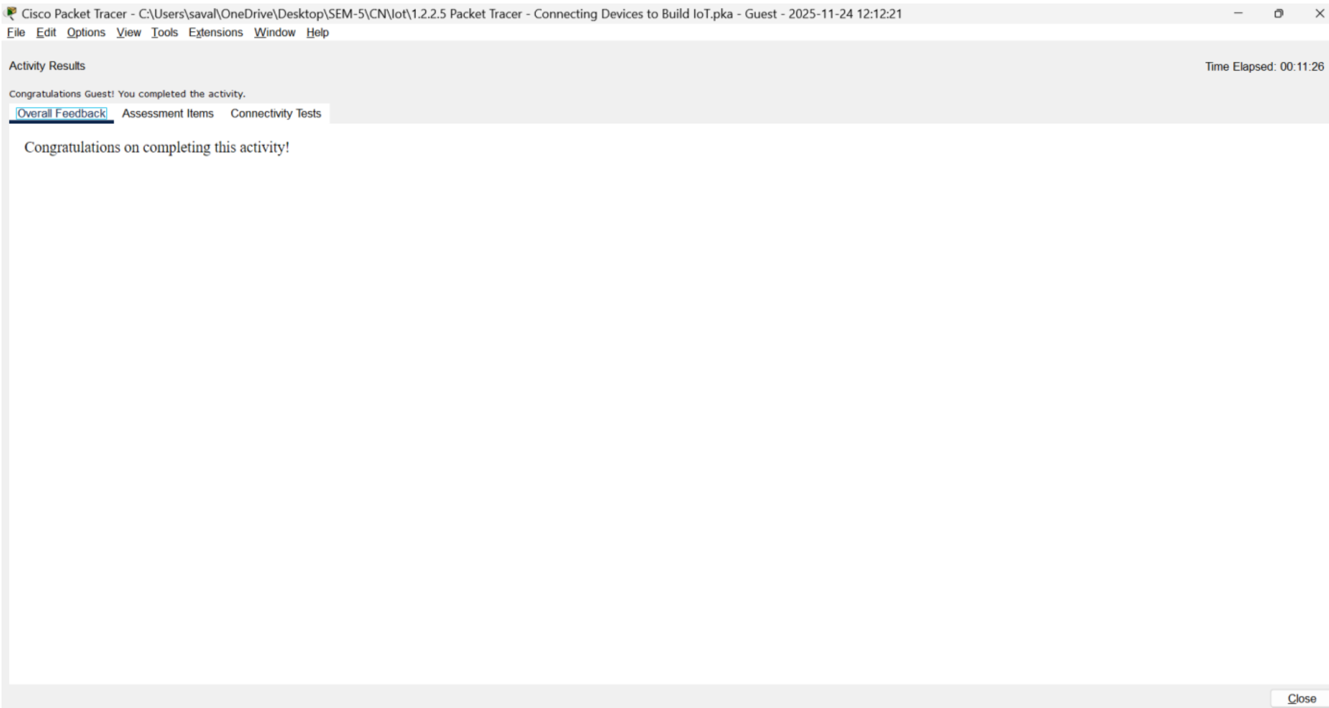
Why are the other devices, the Switch, the Server and the PC not listed? Is this a mistake?

- ➢ Switch, PC and Server are NOT IoT devices, therefore they are not registered with the IoT Server.

g. Click each device to expand it and monitor the status of a specific device.

Part 4 : Reflection

The power of this IoT solution becomes clear when a user can monitor the power consumption of the system not only locally but also remotely. One step further would be to connect a microcontroller and write code to turn off one or more the LEDs when the battery power dropped below a pre-defined threshold. This *energy saving* mode would allow the battery some time to recharge before all the LEDs could be brought up again.



Packet Tracer – Simulating IoT Devices

Objectives

Part 1: Build the Circuit

- Place the components in the Logical Workspace
- Connect the components

Part 2: Program the Single Board Computer (SBC)

- Run the default program
- Modify the default program

Background / Scenario

Packet Tracer has evolved to simulate IoT devices. This tutorial will guide you through the process of placing components in the Logical Workspace, connecting the components, and then programming the single-board computer (SBC) to control them.

Required Resources

- PC with Packet Tracer 7.1 or newer installed

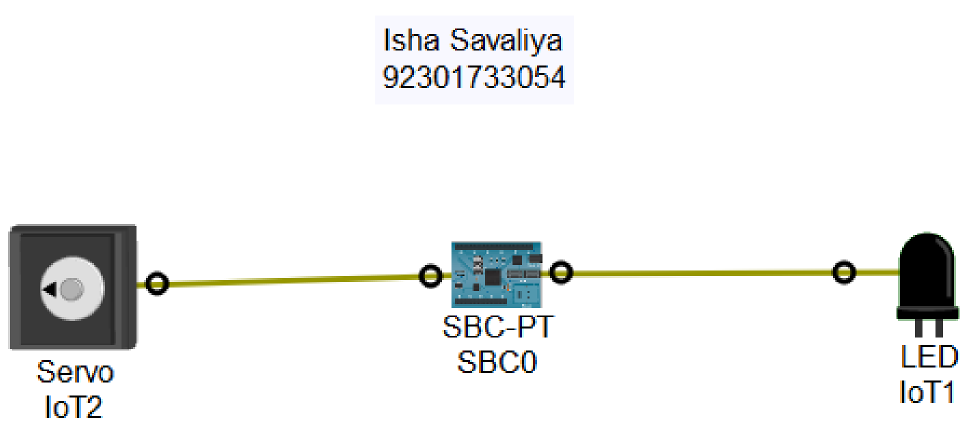
Part 1 : Build the Circuit

Step 1: Place components in the logical workspace.

a. Open Packet Tracer 7.1 or newer, and choose the Components icon.

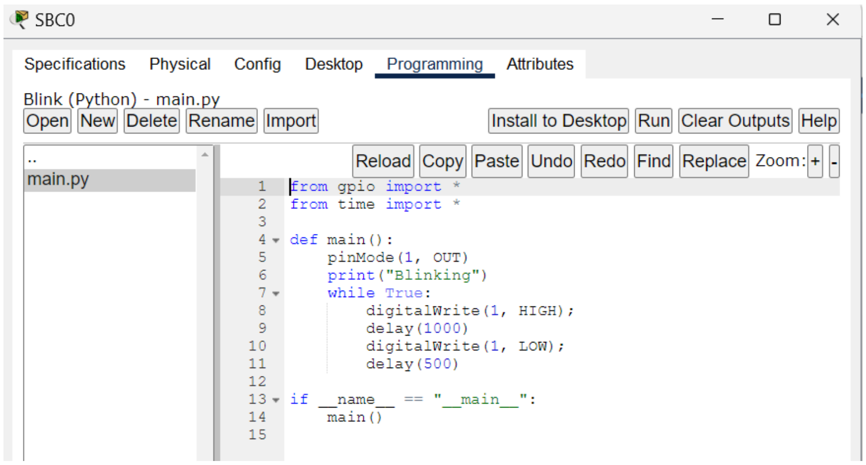
b. Place a SBC Board in the Logical Workspace.
c. Place an LED and a Servo in the Logical Workspace.

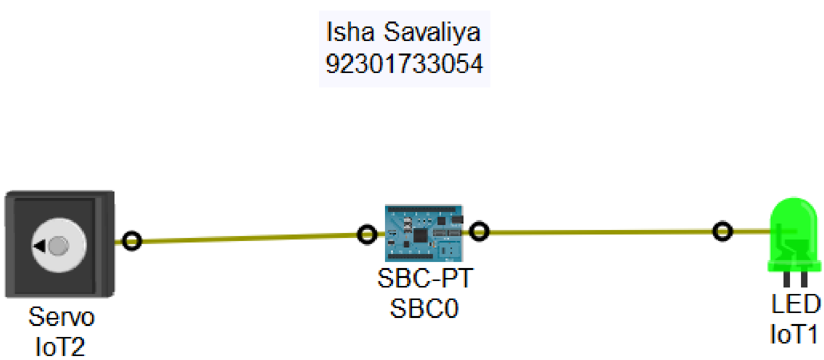
Step 2: Connecting the components.
a. Click the Connections icon, select an IoT Custom Cable, and connect SBC0 D0 to Servo0 D0.
b. Select another IoT Custom Cable and connect SBC0 D1 to LED D0.



Step 1 : Run the default program.
a. Double-click SBC0 and select the Programming tab.
b. Double-click Blink (Python) in the left pane to open it.
c. Double-click main.py to reveal the default Python code.

d. Click the Run button to run the default code. Return to the Logical Workspace. The LED should be blinking.
e. Return to the SBC0 Programming tab, and click the Stop button to stop the program execution.

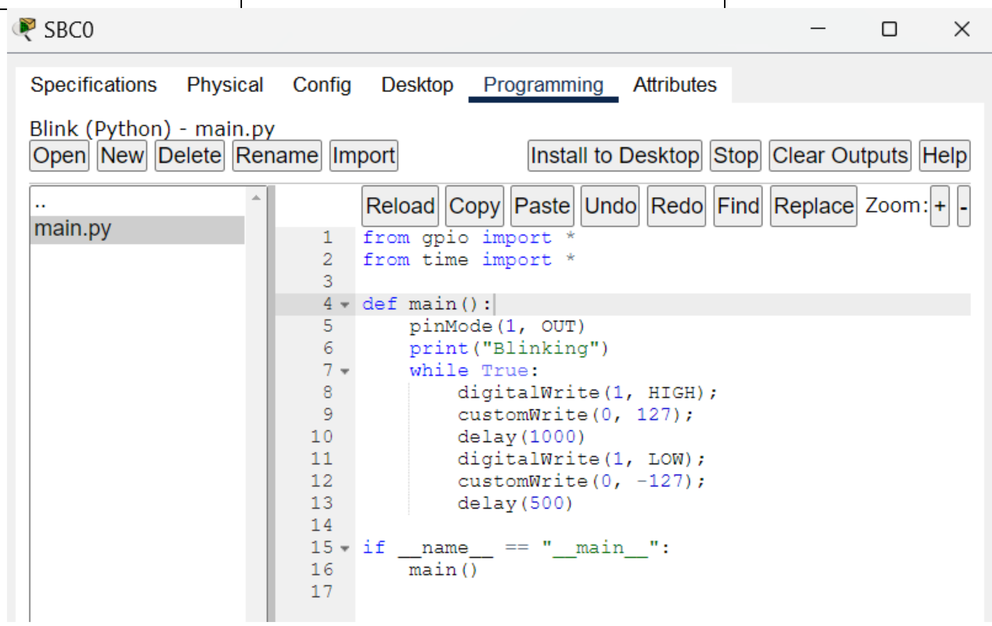


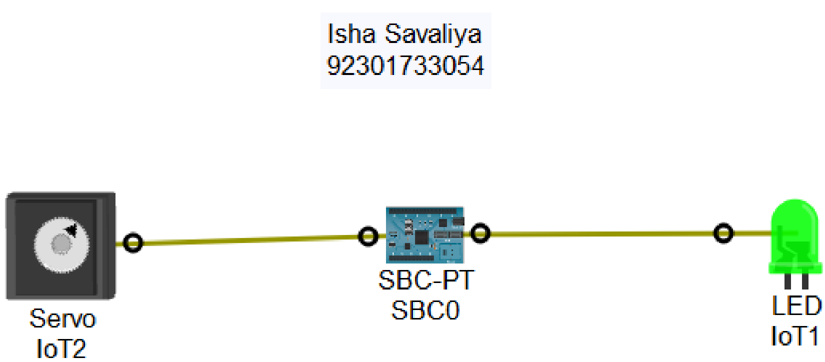


Step 2 : Modify the default program.
a. Copy line 8 of the source code and paste it just below line 8. Do the same with line 11 (formerly line 10) and paste it immediately after the original line of code.
b. Modify the new lines of code to read:
customWrite(0, 127);
and
customWrite(0, -127);
c. Run the modified program. The servo should now move along with the blinking LED.

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Computer Networks (01CT0503)	**Aim: Design and simulate IoT scenario.**

Experiment No: 13	**Date: 23-11-2025**	**Enrolment No: 92301733024**





Reflection

What could be changed to make the servo turn in the opposite direction while the LED is blinking?

- To make the servo turn in the opposite direction while the LED is blinking, I would reverse the values sent to the servo using customWrite, for example changing 127 to -127 and -127 to 127. This would cause the servo to rotate the opposite way while the LED continues blinking.

	Marwadi University **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Computer Networks (01CT0503)	**Aim: Design and simulate IoT scenario.**

Experiment No: 13	**Date: 23-11-2025**	**Enrolment No: 92301733024**

Conclusion:

In this experiment, we successfully designed and simulated an IoT-based solar power system using Cisco Packet Tracer. We connected a Solar Panel, Power Meter and Battery, configured them using DHCP, and registered them to an IoT Server for remote monitoring. We verified power generation, storage and consumption using LEDs, and monitored device status through the IoT web dashboard. This simulation demonstrates how IoT enables real-time monitoring, remote access, and automation in smart energy systems.