Lab2: IoT devices configuration

1. Things and Components available in Packet Tracer

In addition to classical network devices such as routers and switches available in the previous versions, Packet Tracer 7.2 **Components Box** now contains a wide variety of Smart Things and components :

- **Smart Things** are physical objects that can **connect to the Registration Server or Home Gateway through a network interface**. They are separated into 4 subcategories : Home, Smart City, Industrial, and Power Grid.
- **Components** are physical objects that **connect to microcontroller (MCU-PT) or single boarded computers (SBC-PT).** They typically does not have a network interface and rely on the MCU-PT or SBC-PT for network access. These are simple devices that only communicate through their analog or digital slots.

There are three subcategories for Components :

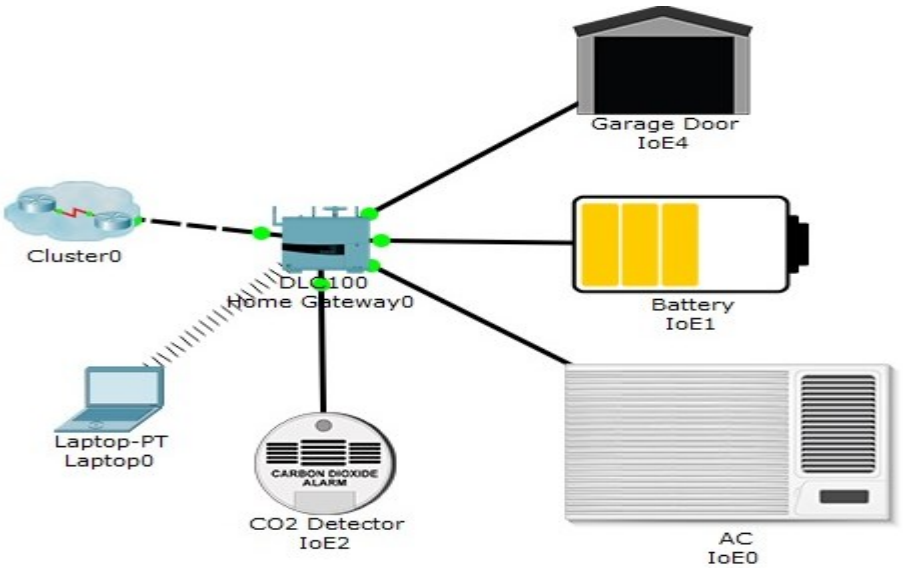
- **Boards**: microcontrollers (MCU-PT), single boarded computers (SBC-PT), and a special device called Thing which are used to create self-contained physical objects like coffee makers or smoke alarms.
- **Actuators**: these components manipulate the Environment, themselves, or the area around them.
- **Sensors**: these components sense the Environment (photo detectors, temperature sensor), the area around them (RFID, metal sensor), or interactions (potentiometer, push button).

MQTT protocol and applications was added in Packet Tracer 7.1 to improve communications between IoT devices

2. IoE registration server

The IoE Things can directly register on a Home Gateway or on a Server device configured with the IoE service.

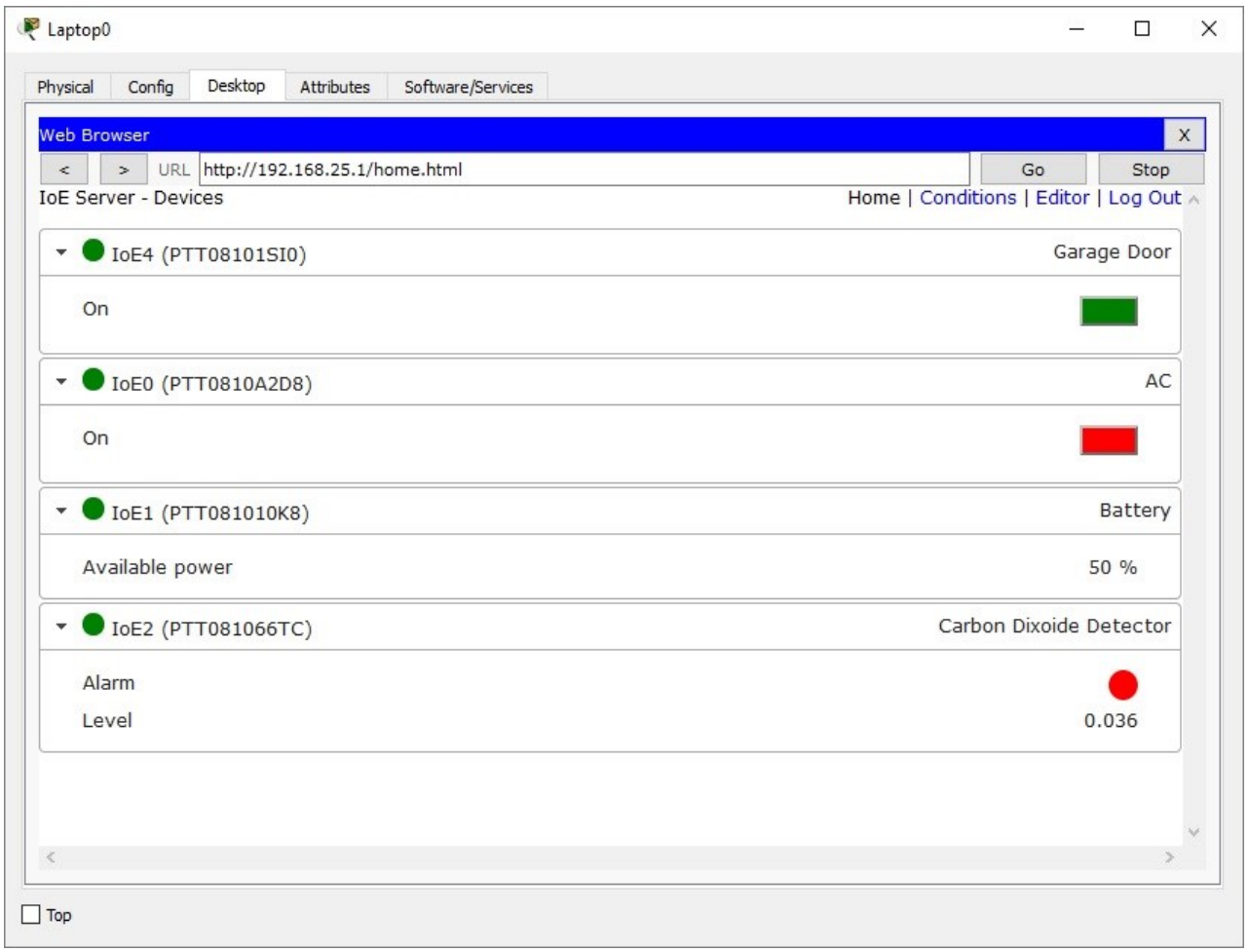
The Home Gateway provides 4 ethernet ports as well as a wireless access point configured with the "HomeGateway" ssid on channel 6. WEP / WPA-PSK / WPA2 enterprise can be configured to secure wireless connections. The picture below shows 4 IOE Things attached to a Home Gateway The Home Gateway is connected to the Internet through it's Internet WAN ethernet port.



The IoE Things can be remotely managed through a web interface hosted by the Home Gateway. The Home Gateway internal (LAN) IP address is 192.168.25.1 but it can also be accessed through it's Internet facing IP address (not configured in this lab, configuration done with a wifi connected laptop). The following screenshot displays the status of the 4 IOE Thngs connected to the gateway.

Home Gateway login (web) : admin

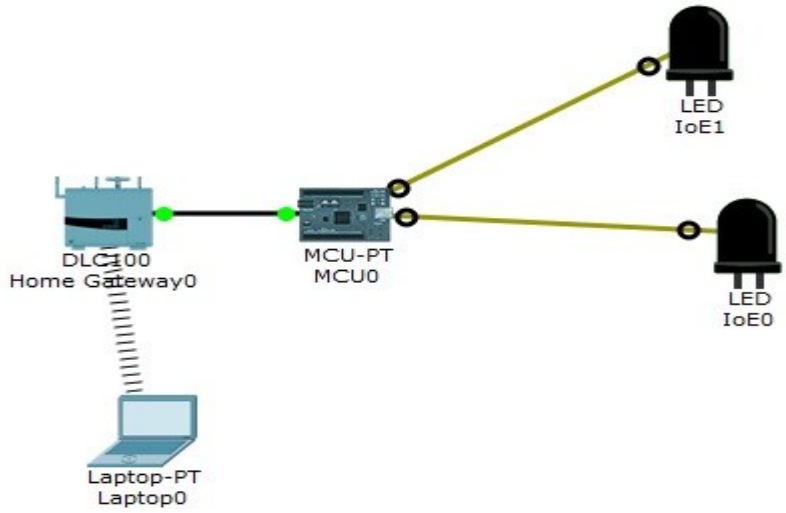
Home Gateway password : admin



IoE components in Packet Tracer 7.1.1

3. Connecting IoE components

As they don't have ethernet interface, IoE components can't be directly connected to the Home Gateway. These smaller sensors or actuators are connected to a microcontrollers (MCU-PT) card. The MCU-PT is connected to the gateway which only sees the microcontroller board but not the IoE components. The Home Gateway relies on the remote control API eventually programmed on the MCU-PT to get the sensor's status.

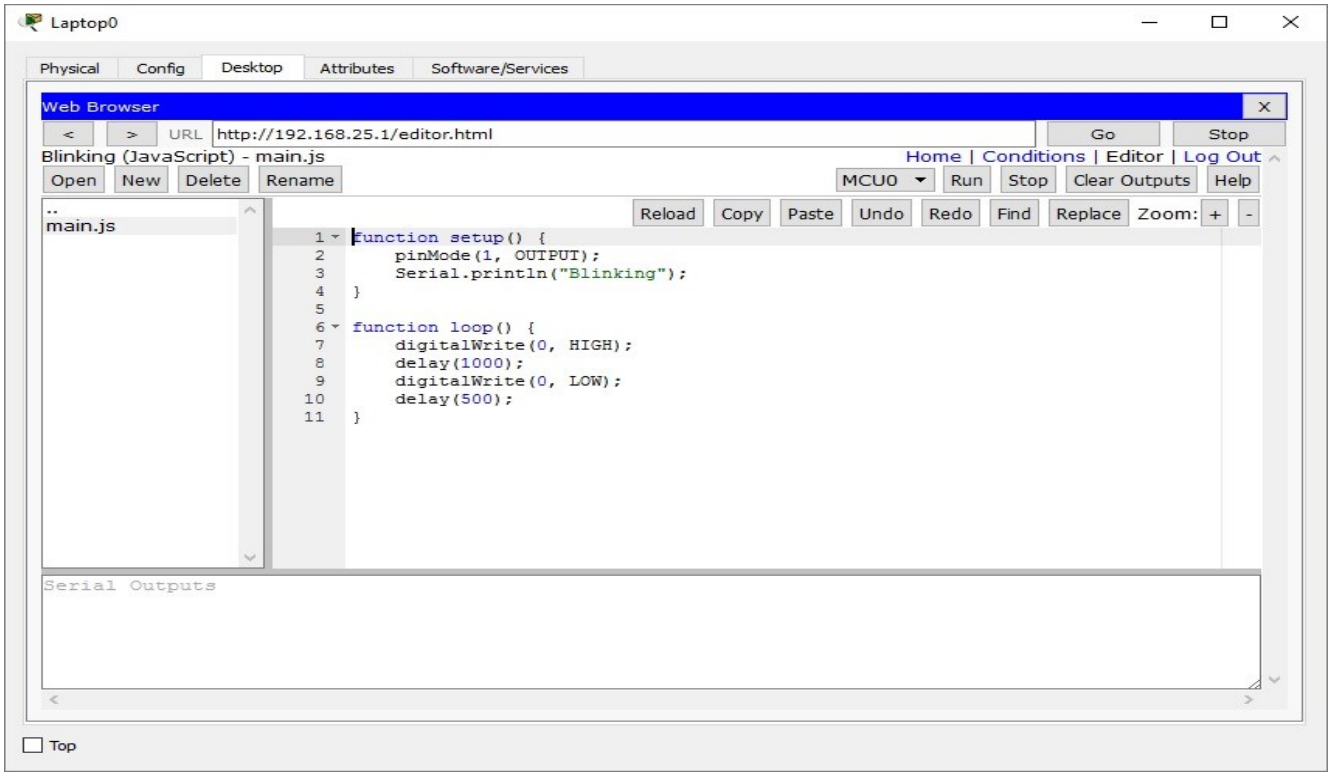


Real world MCU are for example **Arduino boards** like the Arduino Yún Shield

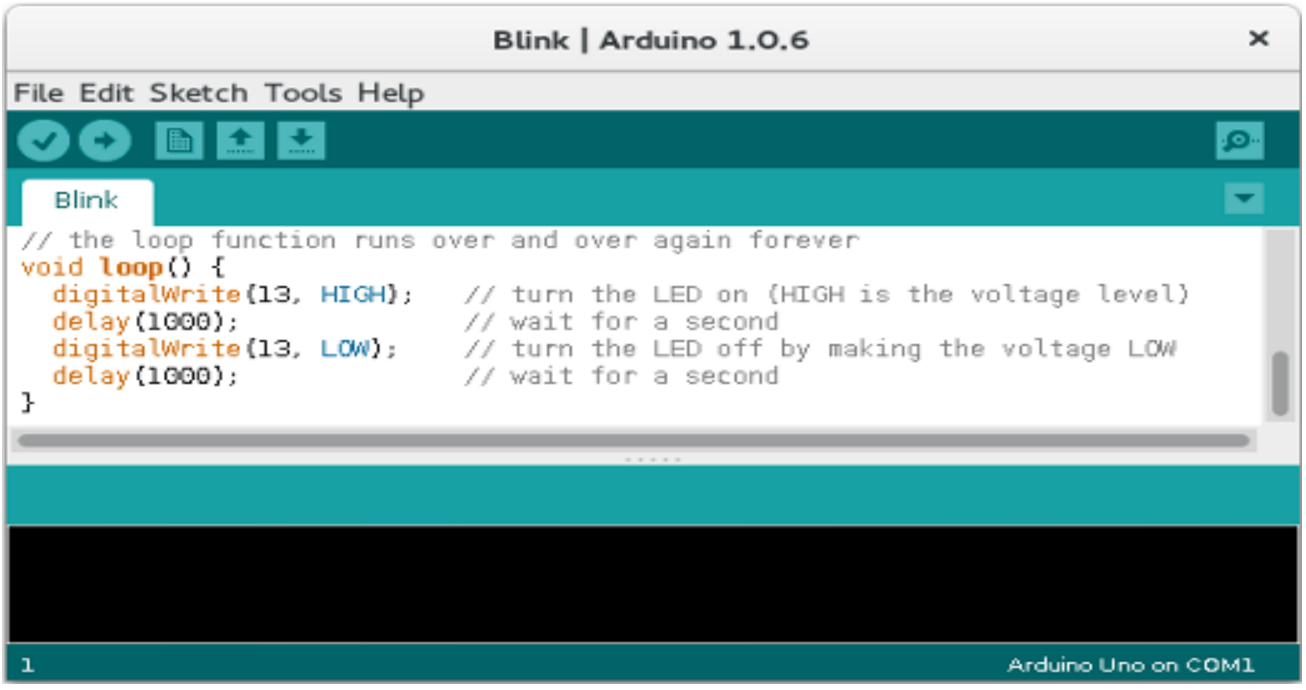
4. Programming IoE components

An IoE programming editor is included in the Home Gateway web interface. It allows Javascript or Python programming of the MCU-PT microcontroller. Code is programmed through the web interface and then published to the MCU board. The following sample code makes the led connected to the digital0 port blink.

```
function setup() {
        pinMode(1, OUTPUT);
        Serial.println("Blinking");
}
function loop() {
        digitalWrite(0, HIGH);
        delay(1000);
        digitalWrite(0, LOW);
        delay(500);
}
```



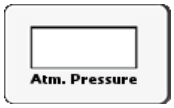
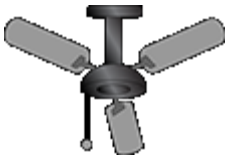
As you can see on the screenshot below, Packet Tracer 7.1.1 emulates Arduino integrated development environment (IDE) for IoT objects programming.



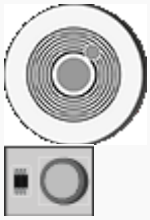
5. Things interacting with Packet Tracer 7.1.1 simulated environment

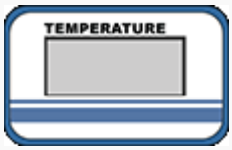
Packet Tracer 7.0 introduced dynamic environent management (temperature, gas, pressure, light, ...) to make IoT device simulation more realistic. Many devices or Things affect or respond to the environment in Packet Tracer : a Fire Sprinkler will raise the water level and humidity in a container, an old car will increase various gases and ambient temperature when turned on, a smoke detector can be used to trigger an alarm when the smoke in environment increases to a certain point...

The table below lists all the things available in Packet Tracer 7.0 and their behaviors with respect to the simulated environment :

Thing	*Icon*	*Environment Behavior*
ATM Pressure Sensor		Detects the Atmospheric Pressure and displays it. The default detection range is from 0 to 110 kPa.
Carbon Dioxide Detector		Detects Carbon Dioxide.
Carbon Monoxide Detector		Detects Carbon Monoxide.
Door		Affects Argon, Carbon Monoxide, Carbon Dioxide, Hydrogen, Helium, Methane, Nitrogen, O2, Ozone, Propane, and Smoke. When the door is opened, those gases will decrease to a maximum of 2% in total change. When the door is opened, the rate of transference for Humidity and Temperature is increased by 25%. The rate of transference for gases is increased by 100%.
Fan		Affects Wind Speed, Humidity, and Ambient Temperature. At Low Speed Setting, the Wind Speed is set to 0.4 kph. The rate of cooling the Ambient Temperature is set to -1C/hour. The rate of reducing the Humidity is set to -1% per hour. At High Speed Setting, the Wind Speed is set to 0.8 kph. The rates of change for Ambient Temperature and Humidity is two times of the low setting.

Fire Sprinkler, Ceiling Sprinkler		Affects Water Level at a rate of 0.1 cm per second. Affects Humidity at a rate of 5% per hour.
Garage Door		Affects Argon, Carbon Monoxide, Carbon Dioxide, Hydrogen, Helium, Methane, Nitrogen, O2, Ozone, Propane, and Smoke. When the door is opened, those gases will decrease to a maximum of 4% in total change. When the door is opened, the rate of transference for Humidity and Temperature is increased by 50%. The rate of transference for gases is increased by 100%.
Home Speaker, Speaker		Affects Sound Volume at 65 dB. Affects Sound Pitch at 20 CPS to 60 CPS. Affects White Noise at 20%.
Humidifier		Affects Humidity at a rate of 1% per hour.
Humidity Sensor		Detects Humidity.
Humiture Monitor, Humiture Sensor		Detects Ambient Temperature and Humidity and outputs the value as a sum of the Ambient Temperature and Humidity divided by 2.
Lawn Sprinkler, Floor Sprinkler		Affects Water Level at a rate of 0.1 cm per second. Affects Humidity at a rate of 5% per hour.

LED		Affects Visible Light with a maximum output of 1%.
Light		Affects Visible Light with a maximum output of 20%.
Old Car		Affects Carbon Monoxide at a rate of 1% per hour. Affects Carbon Dioxide at a rate of 2% per hour. Affects Smoke at a rate of 3% per hour. Affects Ambient Temperature at a rate of 1% per hour.
Photo Sensor		Detects Visible Light.
Piezo Speaker		Affects Sound Volume at 10 dB. Affects Sound Pitch 20 CPS.
RGB LED		Affects Visible Light with a maximum output of 2%.
Smart LED, Dimmable LED		Affects Visible Light with a maximum output of 3%.
Smoke Detector, Smoke Sensor		Detects Smoke.
Solar Panel		Detects Sunlight to generate electricity.

Temperature Monitor		Detects Ambient Temperature.
Temperature Sensor		Detects Ambient Temperature.
Water Level Monitor, Water Detector		Detects Water Level.
Wind Sensor		Detects Wind Speed.
Wind Turbine		Detects Wind Speed to generate electricity.
Window		Affects Argon, Carbon Monoxide, Carbon Dioxide, Hydrogen, Helium, Methane, Nitrogen, O2, Ozone, Propane, and Smoke. When the door is opened, those gases will decrease to a maximum of 1% in total change. When the door is opened, the rate of transference for Humidity and Temperature is increased by 20%. The rate of transference for gases is increased by 100%.
Drain Actuator		Affects Water Level at a rate of -0.5 cm per second.
Sound Frequency Detector		Detects Sound Pitch and displays it.
Furnace, Heating Element		Affects Humidity at a rate of -2% per hour. Affects Ambient Temperature at a rate of 10°C per hour.

AC, Air Cooler		Affects Humidity at a rate of -2% per hour. Affects Ambient Temperature at a rate of -10°C per hour.

6. Arduino emulation for IoT programming

a. What is Arduino ?

Arduino is a hardware and software company building microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices.

The project is based on microcontroller board designs, produced by several vendors, using several microcontrollers. Arduino boards provide digital and analog input/output (I/O) pins that can interface to different expansion boards and other circuits. The boards feature serial and USB communication interfaces for loading programs from computers to the boards.

Programming the microcontrollers is done through the Arduino integrated development environment (IDE) based on a programming language named Processing, which also supports the languages C and C++.

b. Arduino at Cisco Live 2016

Arduino organised a technical challenge at Cisco Live 2016. The idea was that people at the Cisco event could discover and play with some fun Arduino projects like a RFID lock, a colour sensing and display thing, and a light sensitive theremin.

c. Arduino language implemented in Packet Tracer 7.2

Packet Tracer 7.0 was the first Cisco Packet Tracer release including IoT features. This version includes IoT components and microcontroller (MCU-PT) or single boarded computers (SBC-PT) to connect them to the network. The microcontroller boardemulate Arduino hardware and can be programmed using the same Processing language.

Function	**Packet Tracer 7 reference**	**Arduino reference**
Structure		
setup()	If defined, this function is called once when the program starts. function setup() { pinMode(0, INPUT); }	The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board. void setup() {

		Serial.begin(9600);
		pinMode(buttonPin, INPUT);
		}

loop()	If defined, this function is called continuously when the program is running. The frequency of the calls depends on the complexity of this function, the number of other devices running programs and their complexity, and the machine's processing power.
	function loop() {
	Serial.println(digitalRead(0));
	}
		After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.
		void loop()
		{
		if (digitalRead(buttonPin) == HIGH)
		Serial.write('H');
		else
		Serial.write('L');
		delay(1000);
		}

Digital I/O		

pinMode(slot, mode)	Set a digital slot to INPUT or OUTPUT.
	pinMode(1, OUTPUT);
	pinMode(2, INPUT);
		Configures the specified pin to behave either as an input or an output.
		void setup()
		{
		pinMode(ledPin, OUTPUT); // sets the digital pin as output
		}
		void loop()
		{
		digitalWrite(ledPin, HIGH); // sets the LED on
		delay(1000); // waits for a second
		digitalWrite(ledPin, LOW); // sets the LED off

		delay(1000); // waits for a second
		}

digitalRead(slot)	Reads from a digital slot, returns HIGH or LOW.
	var val = digitalRead(1);
		Reads the value from a specified digital pin, either HIGH or LOW
		void loop()
		{
		val = digitalRead(inPin); // read the input pi n
		digitalWrite(ledPin, val); // sets the LED to t he button's value
		}

digitalWrite(slot, value)	Writes to a digital slot with HIGH or LOW.
	digitalWrite(1, HIGH);
		Write a HIGH or a LOW value to a digital pin.
		void loop()
		{
		digitalWrite(ledPin, HIGH); // sets the LED o n
		delay(1000); // waits for a second
		digitalWrite(ledPin, LOW); // sets the LED o ff
		delay(1000); // waits for a second
		}

7. Packet Tracer 7.0 - IoT advanced programming & automation

This tutorial will provide guidelines to simulate a fully automated IoT environment using the capabilities of the new Cisco Packet Tracer 7.0 devices.

A MCU board connected to smoke and temperature sensors will act as a remote Fire Detection Unit. The board will be securely wifi connected to a 829 ISR router.

A SBC board connected to a visual alarm will act as a Fire Alarm System.

Both IoT board will register to a central IoE server which will provide automation capabilities to raise the Fire Alarm if a fire is detected by the Remote Fire Sensor.

ISR829 router wireless configuration with WPA-PSK encryption

1. Setup an IP address on the wlan-ap0 interface for integrated AP managment

2. Connect to the integrated AP (service-module wlan-ap0 session)

3. Configure the SSID with the following settings. Here is configured the SSID, the authentication mode, and the WPA-PSK key.

```
dot11 ssid TEST
 authentication open
 authentication key-management wpa
 wpa-psk ascii 0 AZERTYUIOP
 guest-mode
```

4. Configure the Dot11Radio0 interface with the following settings. Here is configured the encryption mode and the link with the previously configured SSID.

```
interface Dot11Radio0
 no ip address
 bridge-group 1
 encryption mode ciphers aes-ccm
 ssid TEST
```

5. Go back to the router configuration. The data path of the integrated AP is linked to the router's Wlan-GigabitEthernet0 interface (not the AP interface).

6.Configure the Wlan-GigabitEthernet0 interface as an access port in Vlan 1 :

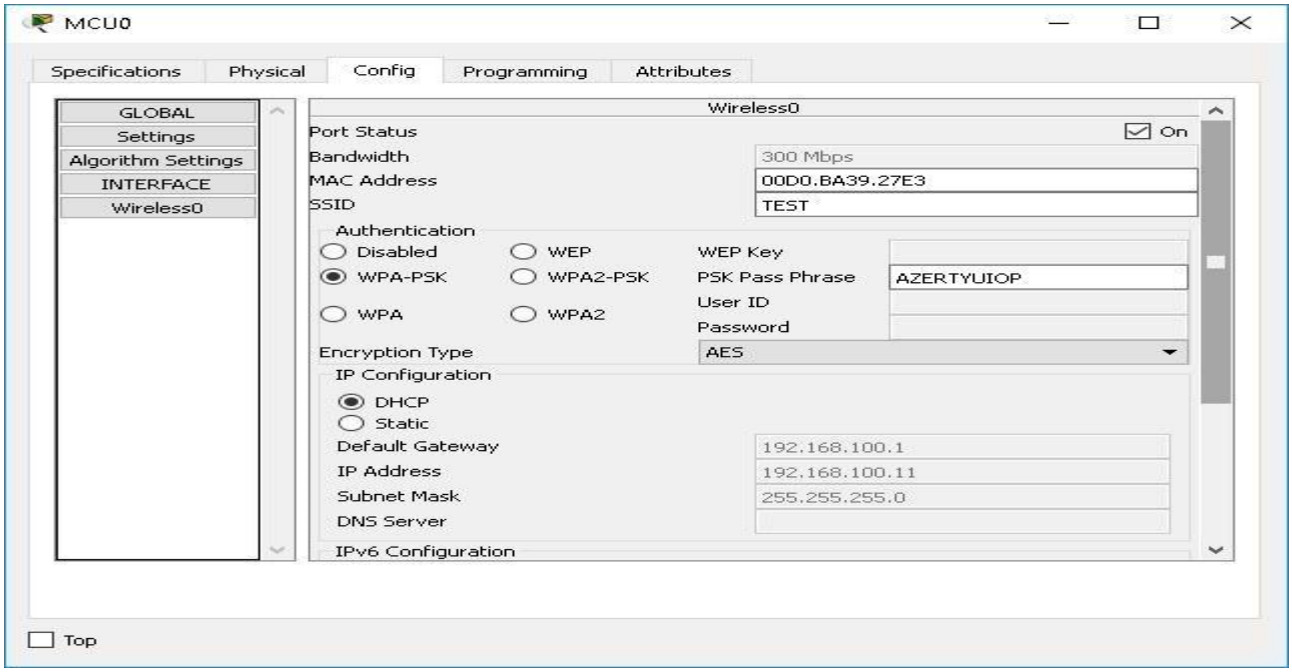
```
interface Wlan-GigabitEthernet0
 switchport mode access
```

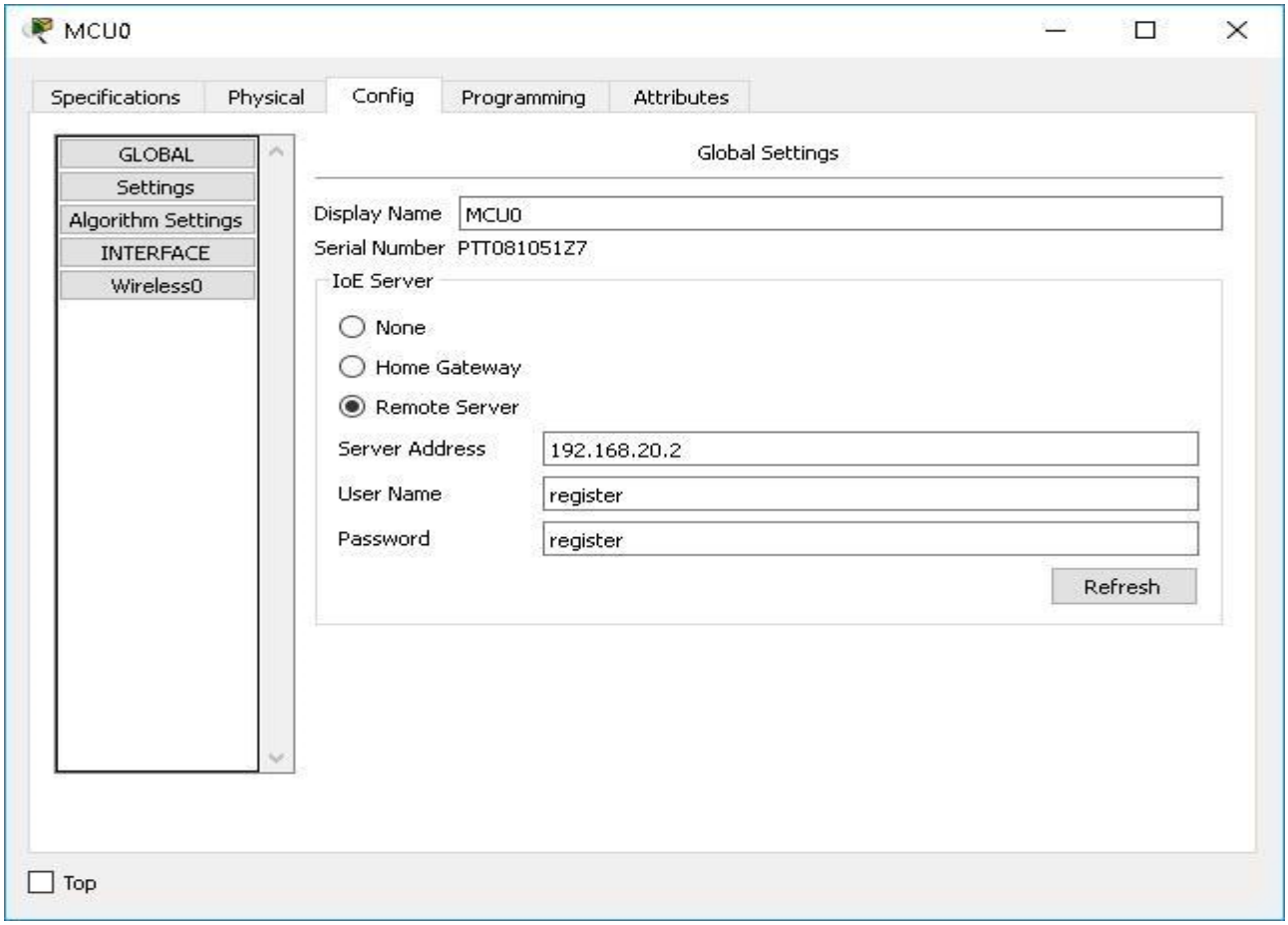
7.Configure DHCP on the router for vlan 1 :

```
ip dhcp excluded-address 192.168.100.1 192.168.100.9
!
ip dhcp pool IoE_POOL
 network 192.168.100.0 255.255.255.0
 default-router 192.168.100.1
 interface Vlan1
 ip address 192.168.100.1 255.255.255.0
```

IoE MCU board registration

Add a wireless connector to the MCU board and configure wireless settings to connect it to the remote 829 router. On the config tab, configure the address of the remote IoE server (192.168.20.2) with the following user/password (user : register, password : register)





Javascript code for the MCU Board (remote fire sensors)

The IoEClient.setup(api) function sets up the API for remote monitor and control from IoE server. The api is an object describing the states reported by the IoE device and identifying the one remote controlable [controlable: true] by the server.

The IoEClient.reportStates(states) reports the states of this device to the IoE server. The argument is a string representing all states of this device. The argument can also be an array representing all states. The number of states must match the length of the states property in the IoEClient.setup() function.

```
function setup() {
            pinMode(1, OUTPUT);
            IoEClient.setup({
             type: "Fire Detection",
             states: [{
                        name: "Fire",
                        type: "bool"
              },
              {
              name: "Temperature",
              type: "number",
              unit: "°C",
              imperialUnit: "°F",
              toImperialConversion: "x*1.8+32",
              toMetricConversion: "(x-32)/1.8",
              decimalDigits: 1
             },
             {
              name: "Smoke Level",
              type: "number",
              unit: "ppm",
              decimalDigits: 1
             }]
            });
}
function loop() {
```

```
        var SmokeLevel = analogRead(A0);
        var Temperature = analogRead(A1);
        var FireDetected=false;
        if (SmokeLevel>50 || Temperature>580) {
        digitalWrite(0, HIGH);
        FireDetected=true;
        }
        else {
        digitalWrite(0, LOW);
        FireDetected=false;
        }
  IoEClient.reportStates([FireDetected,Temperature,SmokeLevel]);
        delay(500);
}
```

Javascript code for the SBC Board (visual fire alarm)

```
var state = 0;
function setup() {
        IoEClient.setup({
                type: "FireAlarm",
                states: [{
                        name: "On",
                        type: "bool",
                        controllable: true
                }]
        });
        IoEClient.onInputReceive = function(input) {
                processData(input, true);
        };
        attachInterrupt(0, function() {
                processData(customRead(0), false);
        });
        setState(state);
```

```
        }
function processData(data, bIsRemote) {
        if ( data.length <= 0  )
                return;
        setState(parseInt(data));
}
function setState(newState) {
        state = newState;
        if ( state === 0 )
                digitalWrite(0, LOW);
        else
                digitalWrite(0, HIGH);
        customWrite(0, state);
        IoEClient.reportStates(state);
        setDeviceProperty(getName(), "state", state);
}
```

8. **Practices - How to Simulate IoT projects using Cisco Packet Tracer**

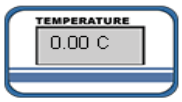
Steps to simulate IoT based Home Automation

a. **Open Packet tracer**

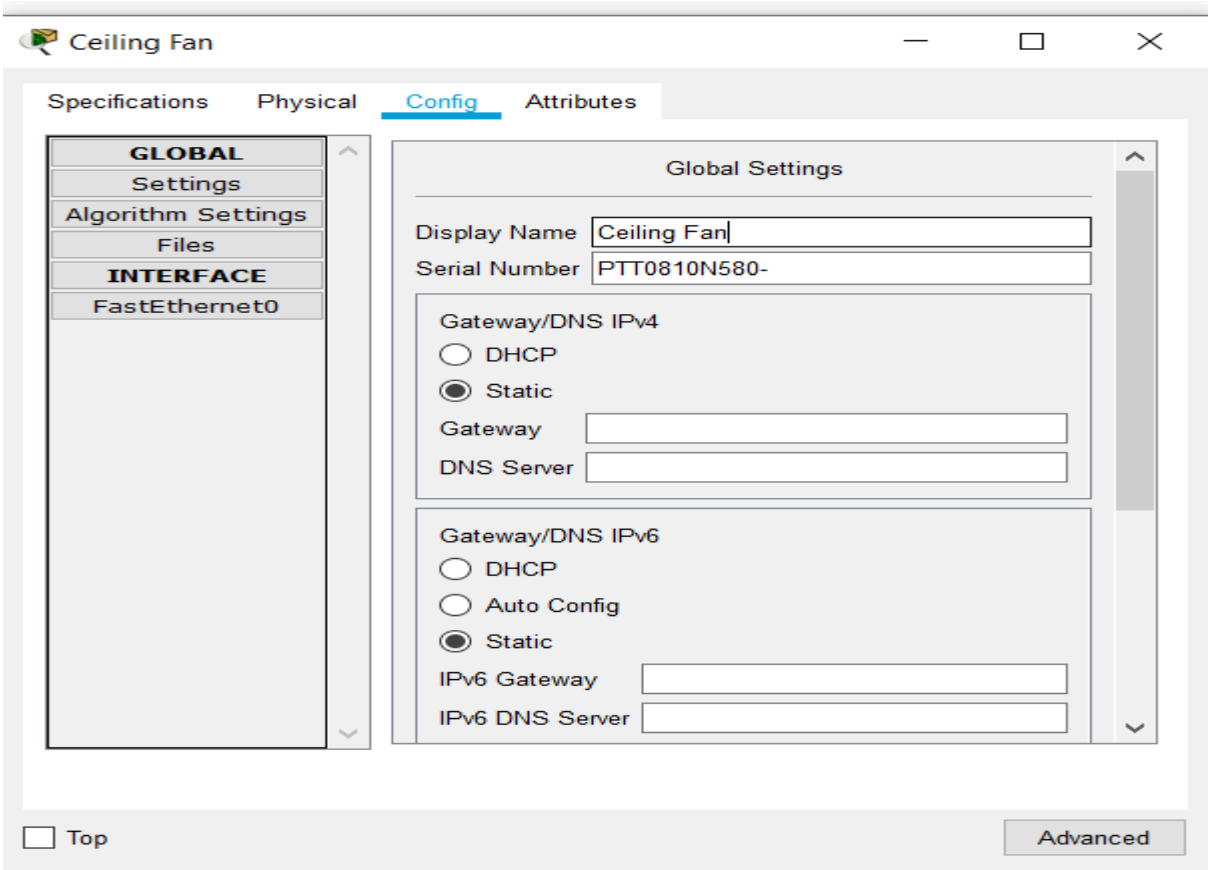
b. Select Home icon at the left bottom corner and now choose the required Home components .

Click on Home icon

C. Drag and place the components

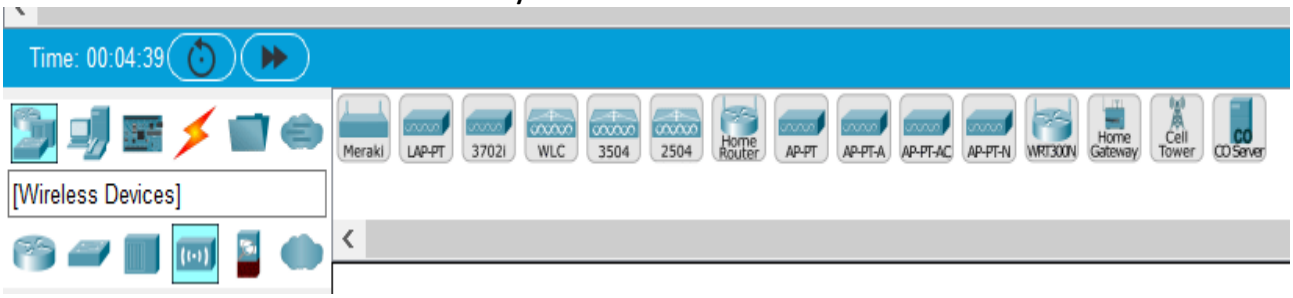


d. Additionally, you can change the name of Appliances by clicking on respective Appliances and select the config option. You have to repeat the same steps to rename the other devices too.



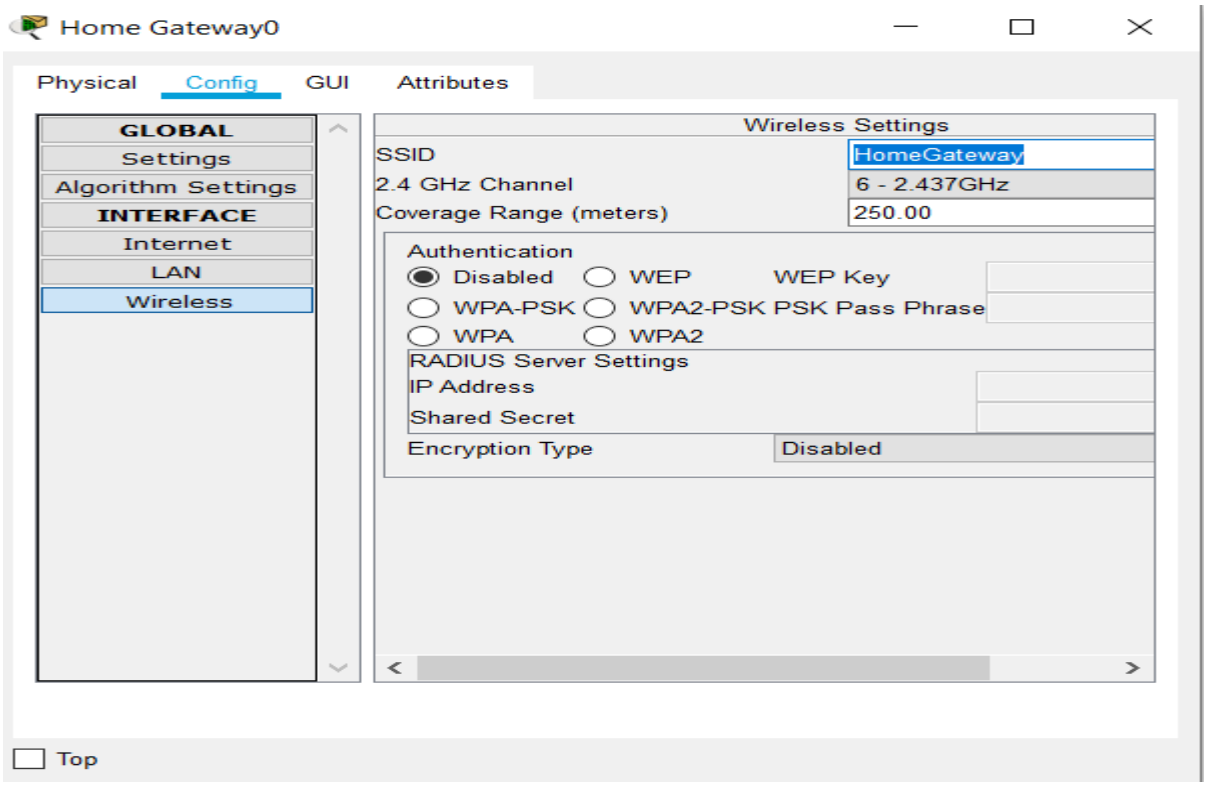
Enter the name

e. Thus now let's establish a wireless connection. Go to Wireless devices and choose Home Gateway



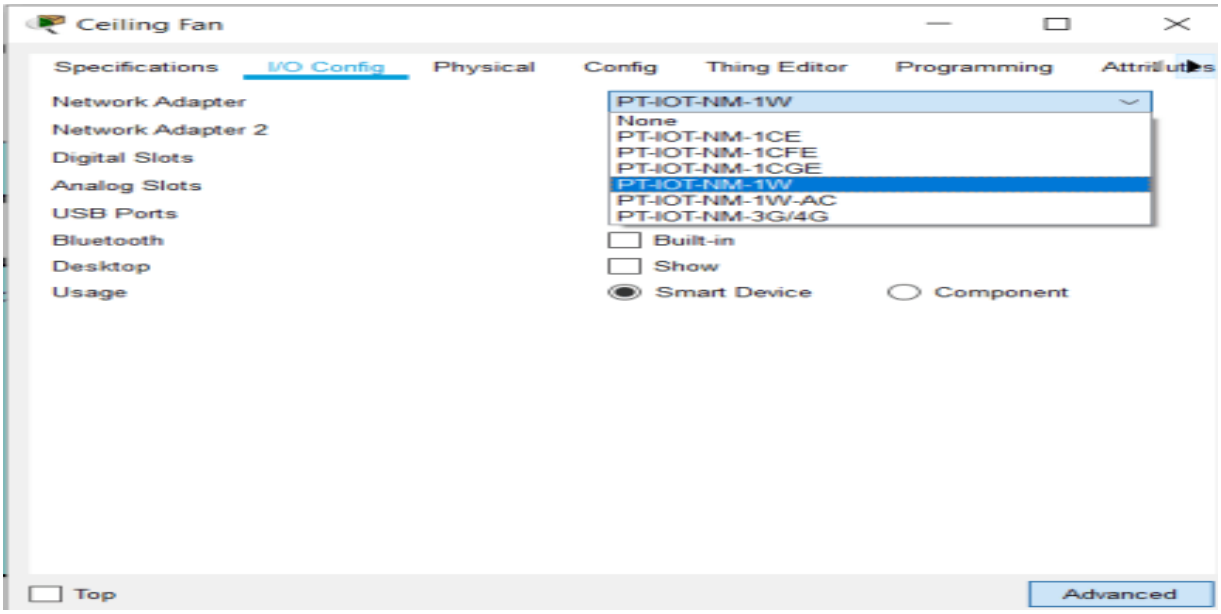
Choose Home Gateway

f. Now copy the SSID of Home Gateway by clicking on the Home gateway.

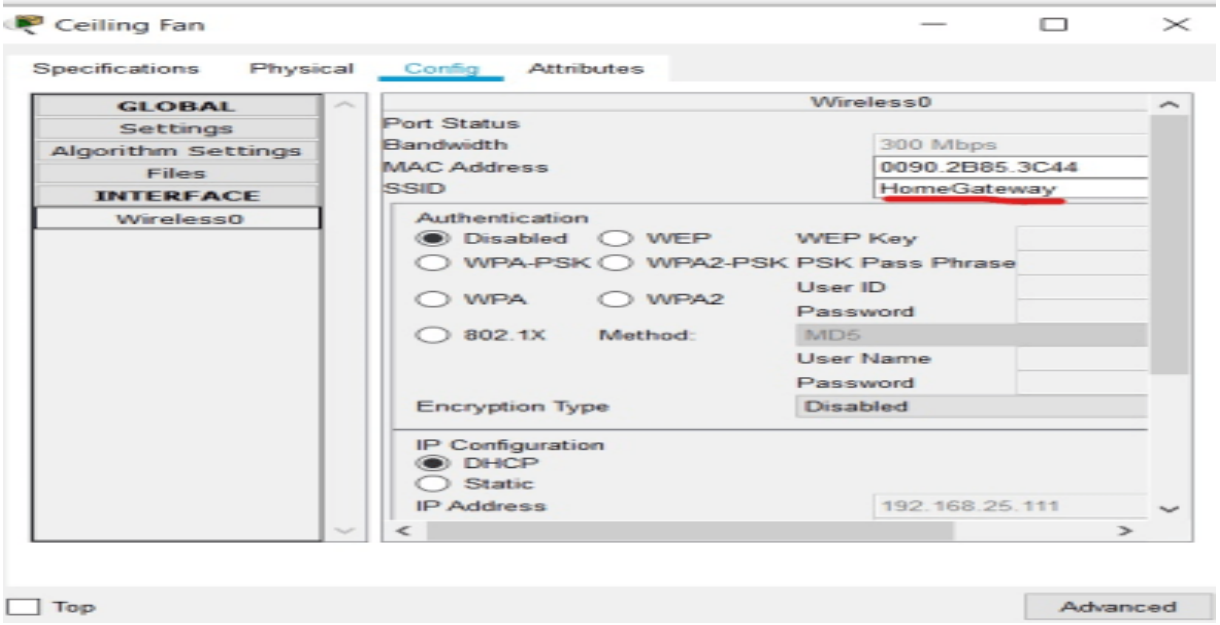


Copy the SSID

g. Therefore It's necessary to configure all devices to connect wirelessly to Home Gateway. Hence you can achieve this by Clicking a device → Advanced →I/O config → select PT-IOT-NW-1W. Similarly, repeat the steps for other devices as well.



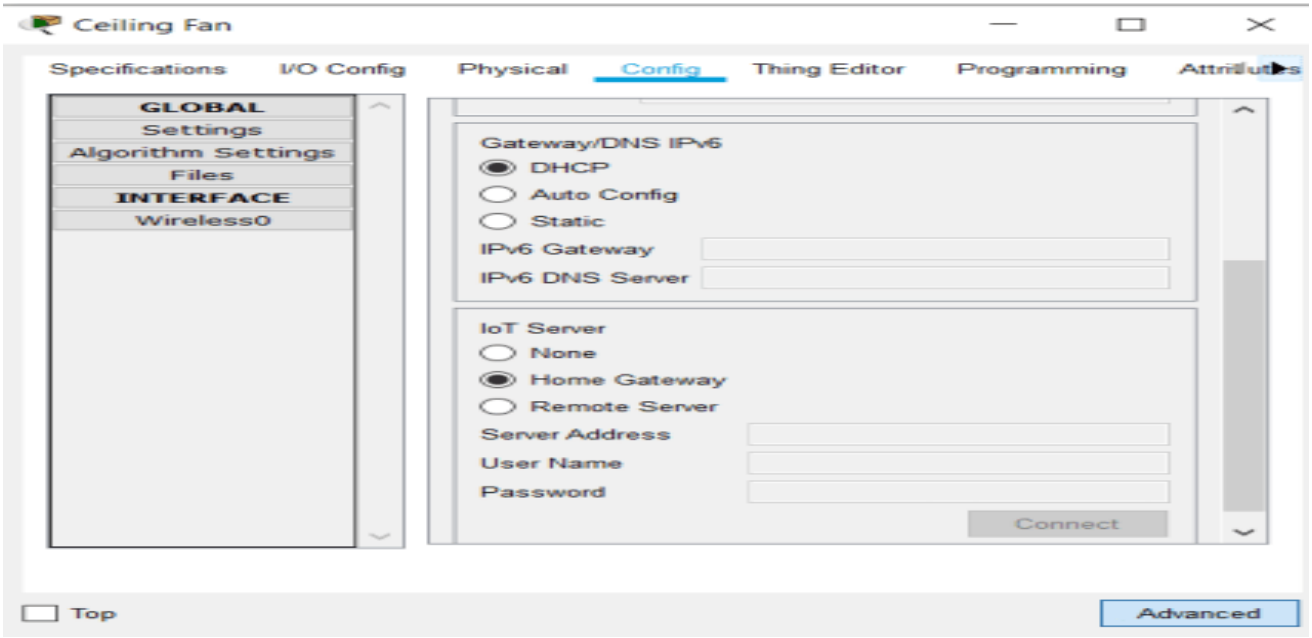
h. Then paste the SSID In each appliance by clicking on the config option. Similarly repeat the steps for other devices as well.



Paste the SSID in all appliances

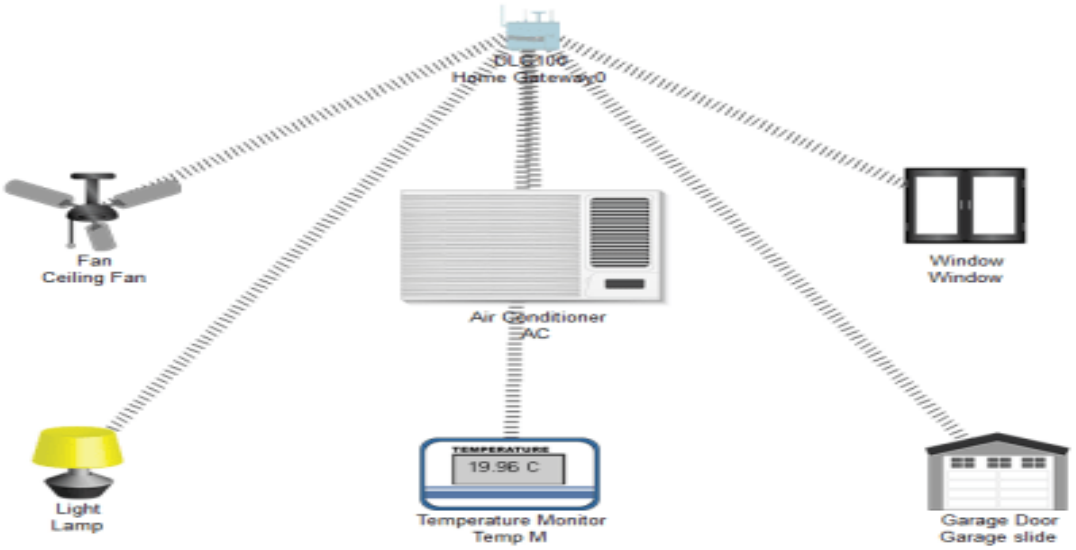
i. Now lets select IOT server as Home gateway to all appliances.

Click on **Appliances →select Advanced → config →choose Home Gateway**. Similarly repeat the steps for other devices as well.



Choose Home Gateway as IOT Server

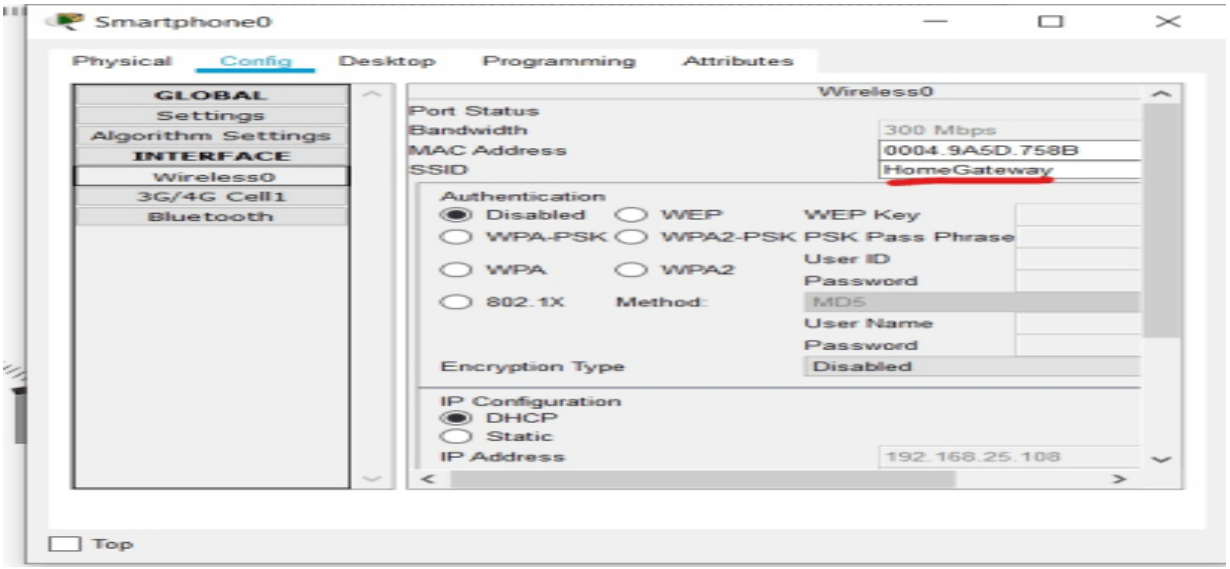
j. Thus you can notice that all devices are connected to the Home Gateway



k. Now select smart device to Control and Monitor Appliances.

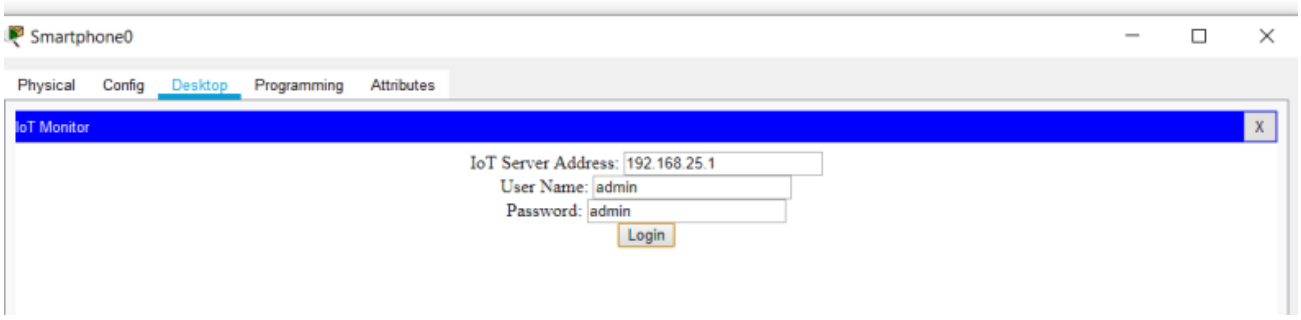
Select end devices at bottom left corner

l. Now connect your smart device also to your Home gateway

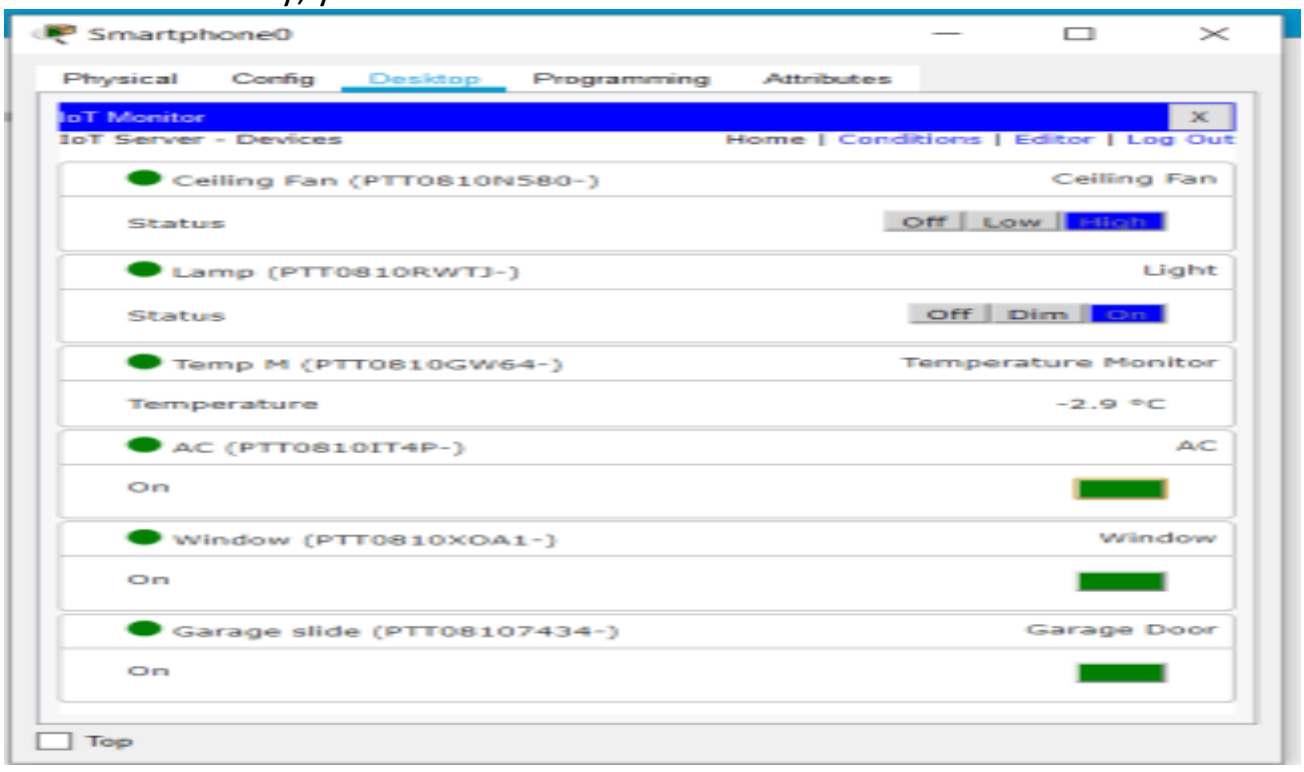


m. Thus your smart device is connected to your Home Gateway.

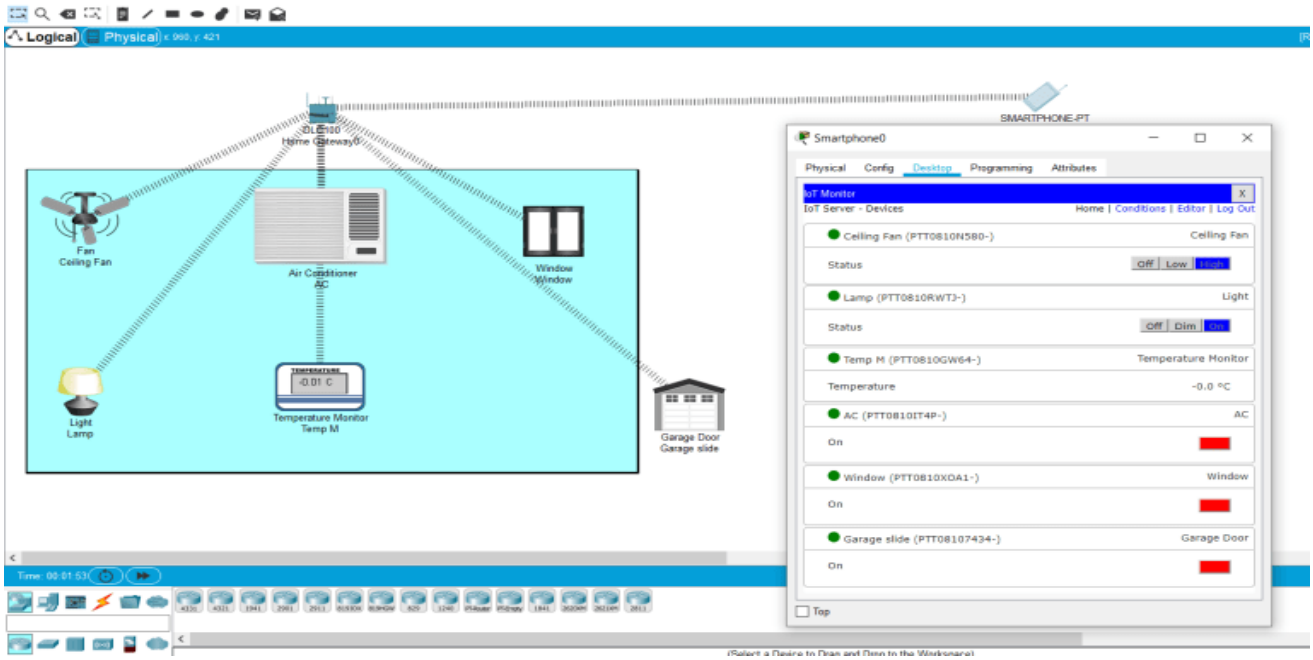
Click on smart device and choose Desktop. You can now login



n. Finally, you can control and Monitor in the Dashboard.



Result


Logical
Physical
SMARTPHONE-PT
Smartphone0
Physical
Config
Desktop
Programming
Attributes
IoT Monitor
IoT Server - Devices
Home | Conditions | Editor | Log Out
Ceiling Fan (PTT0810N580-)
Ceiling Fan
Status
Off
Low
High
Lamp (PTT0810RWT3-)
Light
Status
Off
Dim
On
Temp M (PTT0810GW64-)
Temperature Monitor
Temperature
-0.0 °C
AC (PTT0810IT4P-)
AC
On
Window (PTT0810XOA1-)
Window
On
Garage slide (PTT08107434-)
Garage Door
On
Top
Fan
Ceiling Fan
Air Conditioner
AC
Window
Window
Light
Lamp
Temperature Monitor
Temp M
Garage Door
Garage slide