Lab no 01 – Sensors and Actuators

The purpose of this Lab is to be familiar with sensors and actuators and to learn how to connect them (things). You will use the Arduino board and Arduino IDE.

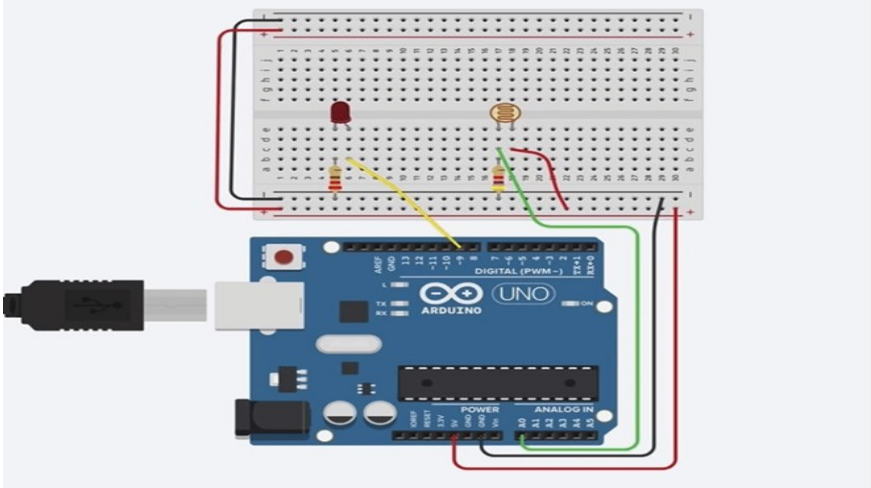
Parts: -

1. Experiment – photoresistor, and LED
2. Sensors and actuators.
3. Closed-loop and Open-loop Control System.

© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.

Part.1 Experiment – photoresistor, and LED

Topology



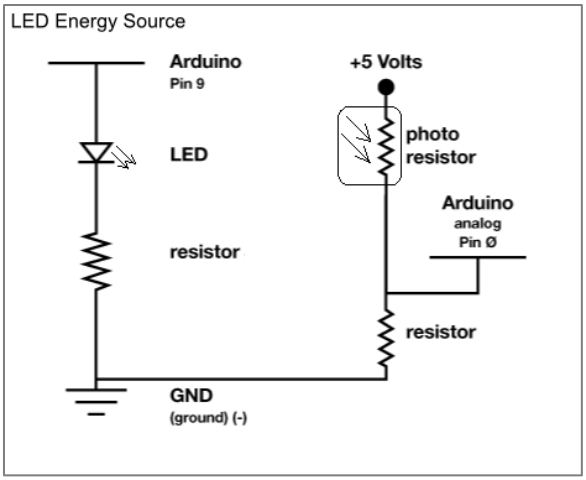
Objectives

- **Configure the Photo Resistor circuit**
- **Load and run the sketch in the Arduino IDE**
- **Modify the sketch to adjust the light range**
- **Convert the sketch to make the circuit work as a night light**

Background / Scenario

This lab will make use of a photoresistor to vary the voltage supplied to an LED based on the value read from the photoresistor. The pins that are used on the Arduino board are noted on the diagram, Pin 9 is used to vary the voltage to the LED which changes the brightness. Pin 9 is capable of simulating analog output using Pulse Wave Modulation (PWM).

© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.



Required Resources

- Arduino Uno Kit or equivalent components.
- USB cable, breadboard, an LED, resistors (220 ohm and 4.7k ohm), photoresistor, and breadboard wires.
- PC configured with Arduino drivers and Arduino software.
- SIK Guide Code example files.

Step 1:Configure the Photo Resistor circuit.

1. Drag an Arduino Uno and breadboard from the components panel to the workplace, next to the existing circuit.

2. Connect breadboard power (+) and ground (-) rails to Arduino 5V and ground (GND), respectively, by clicking to create wires.

3. Plug the LED into two different breadboard rows so that the cathode (negative, shorter leg) connects to one leg of a resistor of 220 ohms. The resistor can go in orientation because resistors aren't polarized, unlike LEDs, which must be connected in a certain way to function.

4. Connect the other resistor leg to the ground.

5. Wire up the LED anode (positive, longer leg) to Arduino pin 9.

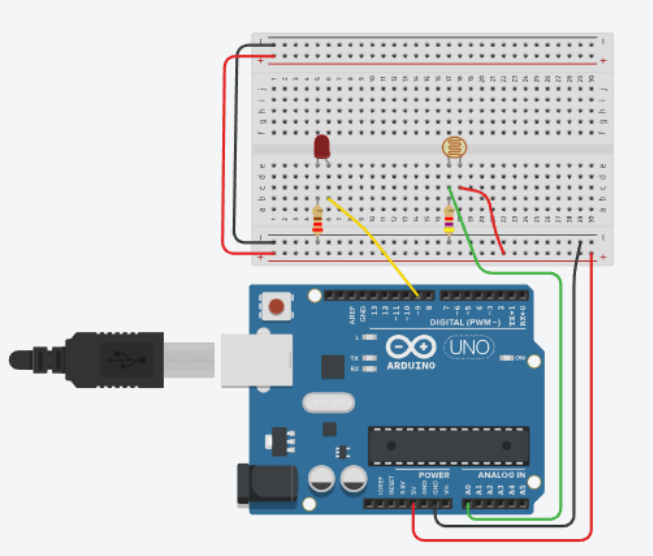
6. Drag a photoresistor from the components panel to your breadboard, so its legs plug into two different rows.

7. Click to create a wire connecting one photoresistor leg to power.

© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.

8. Connect the other leg to Arduino analog pin A0.

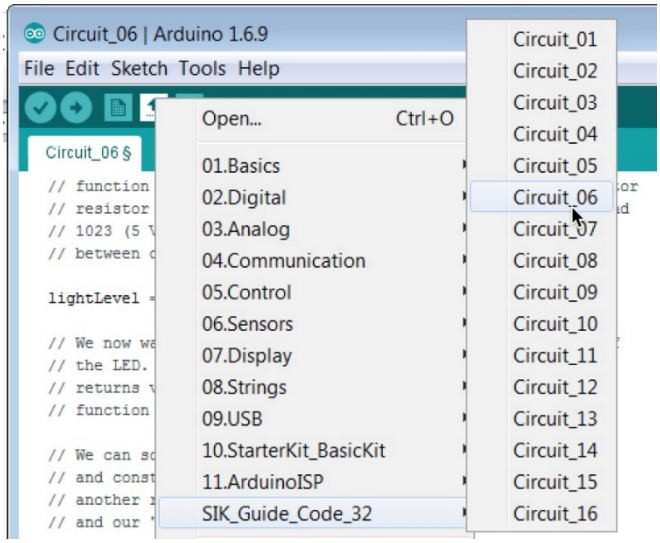
9. Drag a resistor from the components panel to connect the photoresistor leg connected to A0 with the ground and adjust its value to 4.7k ohms.



Load and run the sketch in the Arduino IDE.

a. The Arduino IDE has shortcut buttons directly below the Menu bar. The icon with an up arrow provides a shortcut to the example sketch available. Click the **up arrow** and select **Circuit_06** from the **SIK_Guide_Code_32** folder displayed.

   Note: The **SIK_Guide_Code_32** folder will only exist if the code has been downloaded from http://sparkfun.com/sikcode and copied to the examples folder located under the Arduino program files, typically **C:\Program Files\Arduino\examples**.

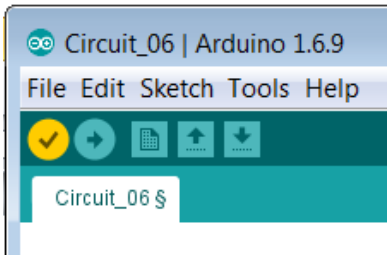


© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.

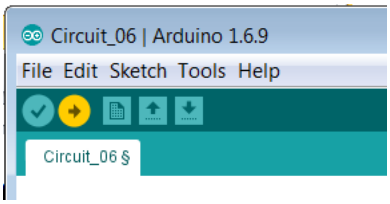
```
/*
SparkFun Inventor's Kit
Circuit 1C-Photoresistor
Use a photoresistor to monitor how bright a room is, and turn an LED on when it gets dark.
This sketch was written by SparkFun Electronics, with lots of help from the Arduino community.
This code is completely free for any use.
View circuit diagram and instructions at: https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40
Download drawings and code at: https://github.com/sparkfun/SIK-Guide-Code
*/
int photoresistor = 0;          //this variable will hold a value based on the brightness of the ambient light
int threshold = 750;            //if the photoresistor reading is below this value the the light will turn on

void setup()
{
  Serial.begin(9600);           //start a serial connection with the computer
  pinMode(13, OUTPUT);          //set pin 13 as an output that can be set to HIGH or LOW
}
void loop()
{
  //read the brightness of the ambient light
  photoresistor = analogRead(A0);   //set photoresistor to a number between 0 and 1023 based on how bright the ambient light is
  Serial.println(photoresistor);    //print the value of photoresistor in the serial monitor on the computer

  //if the photoresistor value is below the threshold turn the light on, otherwise turn it off
  if (photoresistor < threshold){
    digitalWrite(13, HIGH);         // Turn on the LED
  } else{
    digitalWrite(13, LOW);          // Turn off the LED
  }

  delay(100);                   //short delay to make the printout easier to read
}
```

© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.

b. Verify and compile the sketch by clicking the **Checkmark** in the shortcut bar. This creates an executable firmware that can be uploaded to the Flash memory of the microcontroller.



c. Make sure the Arduino is connected to the PC with the USB cable. Upload the firmware by clicking the **Right Arrow** in the shortcut bar.



Note: The LED should light up when the sketch is delivered. The brightness will vary somewhat depending on the lighting conditions in the room.

d. Reduce the amount of light to the photoresistor. One suggestion is to carefully place something like a marker cap down over the photo sensor. The light should dim, it is unlikely to go completely off.

Step 2:Convert the sketch to make the circuit work as a night light.

a. Locate the line of code that reads:

```
analogWrite(ledPin, lightLevel);
```

b. Change the line to read:

```
analogWrite(ledPin, 255-lightLevel);
```

c. Recompile the sketch again by clicking the **Checkmark** in the shortcut bar.

d. Upload the new firmware by clicking the **Right Arrow** in the shortcut bar.

Note: Light levels range from 0 – 255. By subtracting the amount of light read from 255 when it is dark the light level read is 0, therefore the level being written will be 255 or bright light. When there is a lot of light in a room the light will be dim or possibly even off. Try shining a flashlight on the photoresistor to cause the light to go out.

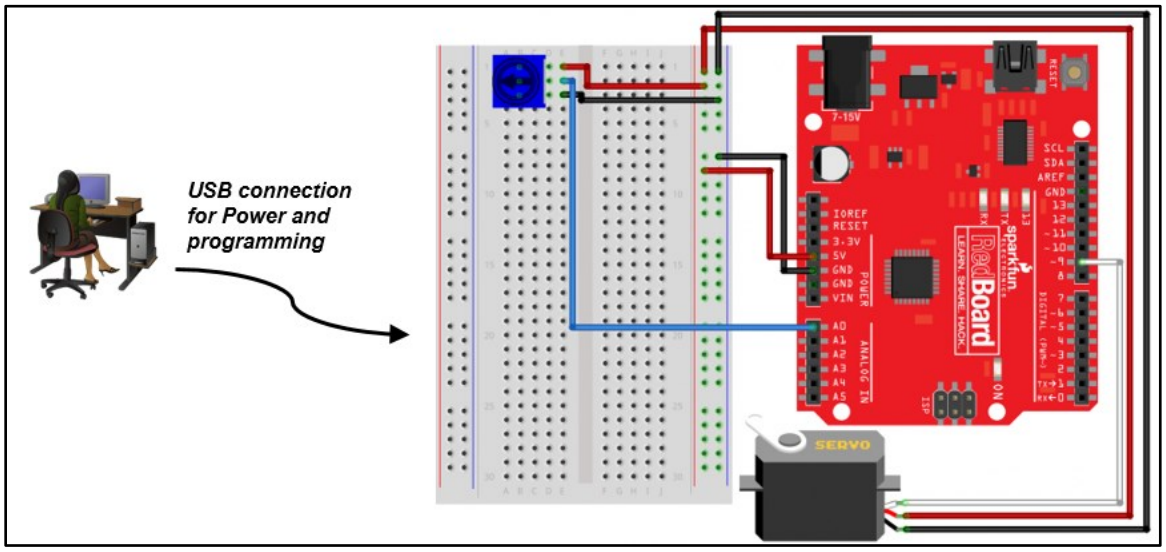
© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.

Part.2 Sensors and Actuators.

Here, we give other examples to connect things, you can apply the same steps above to load and run the sketches.

a) Servo Motor

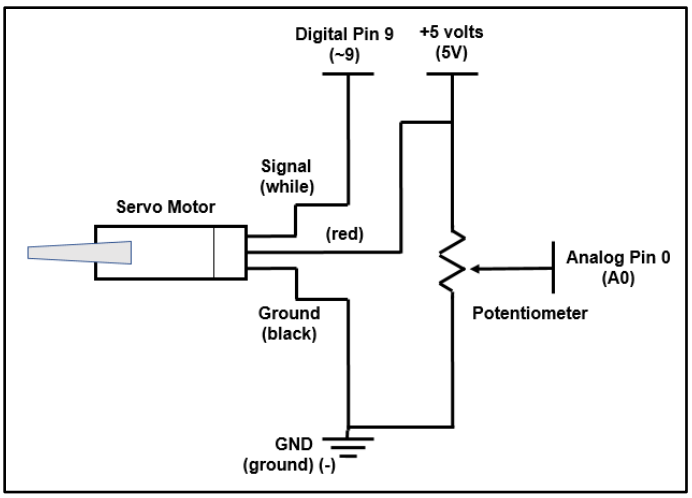
Topology



Background / Scenario

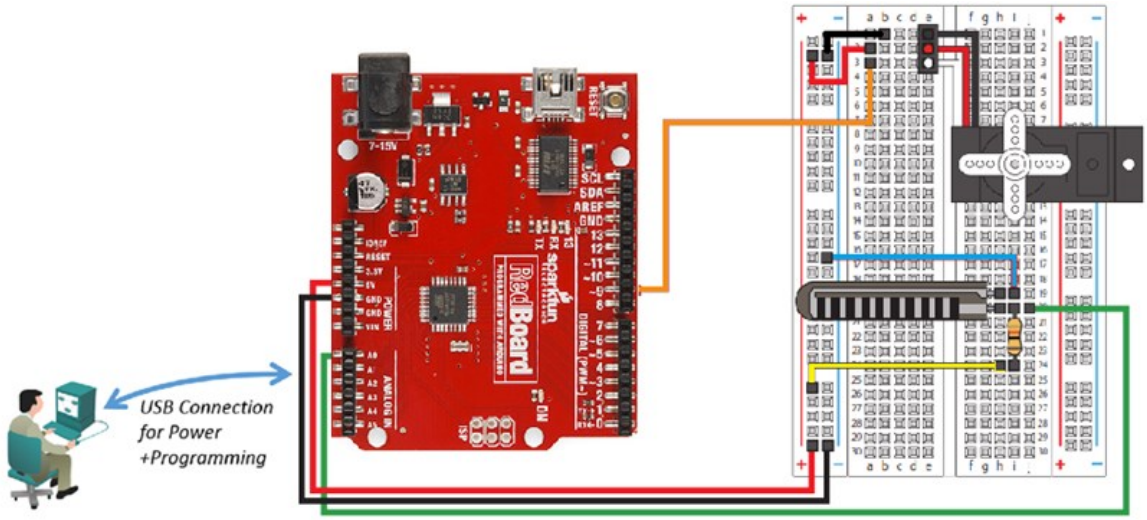
This lab will make use of a potentiometer to change resistance based on the position of the knob on the potentiometer. The varying resistance will then be used to control the voltage sent to the Servo. Varying the voltage will cause the Servo to change positions.

© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.



b) Flex Sensor and Servo

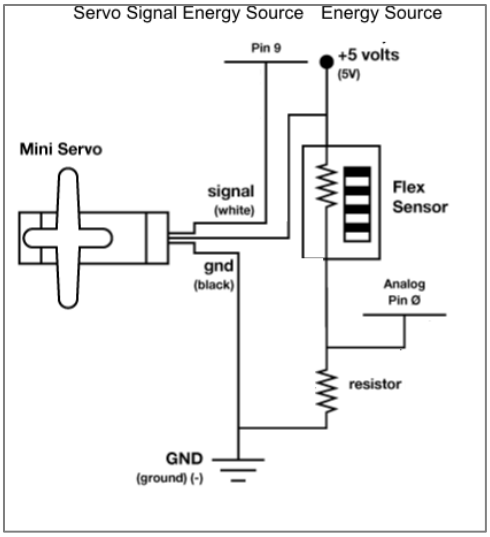
Topology



Background / Scenario

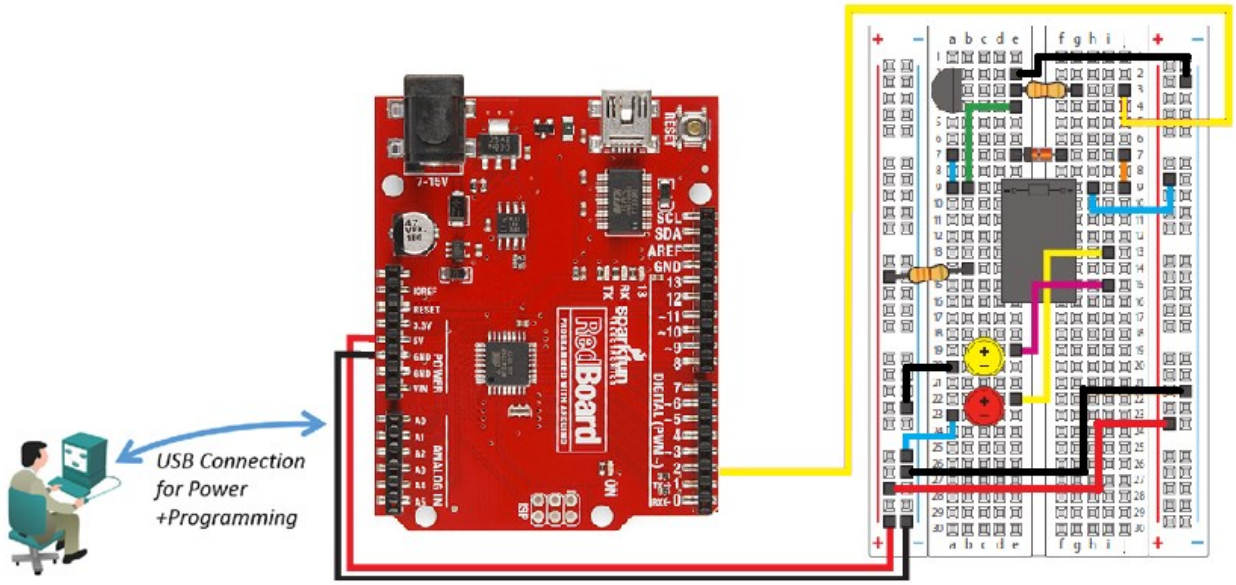
This lab will make use of a special Flex Sensor to change resistance based on the amount of bend placed on the sensor. The varying resistance will then be used to control the voltage sent to the Servo which will cause it to change positions.

© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.



c) Relays

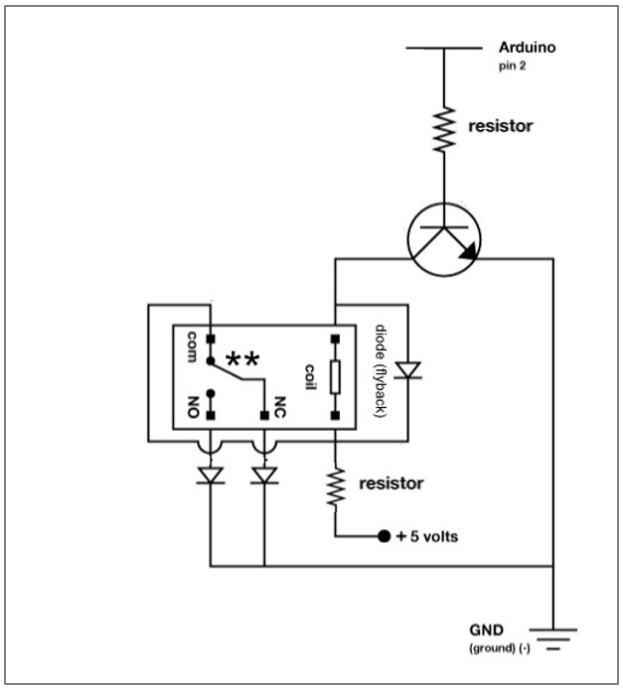
Topology



Background / Scenario

This lab will make use of a relay along with other components to serve as an electrically controlled mechanical switch.

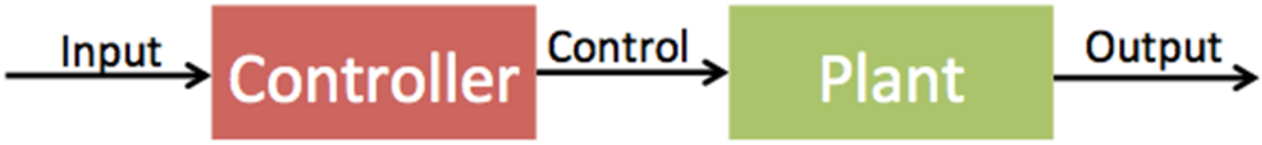
© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.



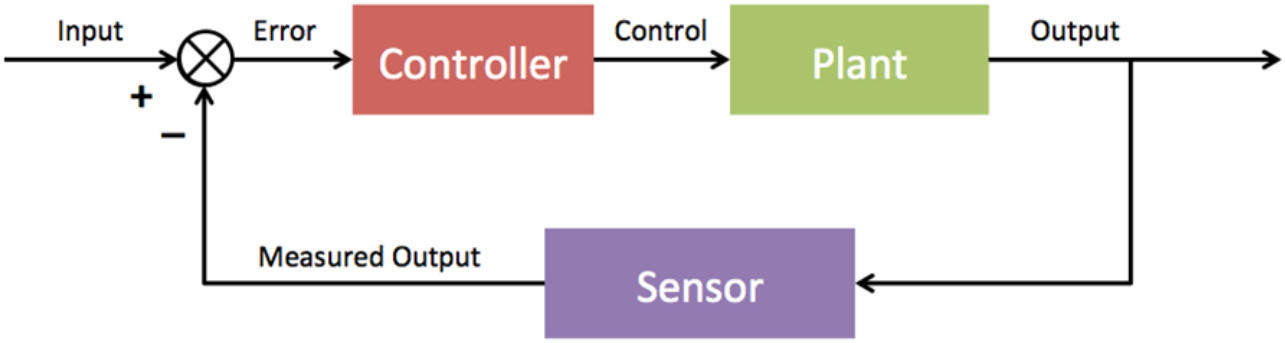
© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.

Part.3 Closed-Loop and Open-Loop Control Systems.

An **open-loop control** system does not monitor the output to determine what adjustments to make to the input. For example, when using a clothes dryer, you might set the timer on the dryer to run the drying cycle for one hour. At the end of the hour, the dryer will stop. The level of dryness of the clothes will vary depending upon their level of wetness at the beginning of the cycle.



In a **closed-loop** control system, the output is measured to determine whether it is the desired output and adjust the input as appropriate. For example, if the clothes dryer is equipped with moisture sensors, the input may be a level of dryness that adjusts the cycle by extending the drying time until the sensors indicate the clothes are dried to the desired level.



© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.

Exercise

Systems	Open/Closed Loop	Explain Your Answer
Clothes dryer with a moisture sensor	Closed	The moisture sensors provide measured output that is compared to a reference for desired dryness. The controller will extend the drying cycle until the input and output are equal.
Light switch to turn on the light		
Thermostat to keep a constant temperature		
Volume on a radio		
Vehicle ignition switch		
Ceiling fan		
Clothes washing machine with water level sensor		

© 2022 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public.