National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member:Umar Farooq **Dated: 12-05-2022**

Semester: 8th **Section: D**

EE-357 Computer and Communication Networks

Open Ended Lab

Arduino Cloud Service For IOT Applications

		PLO5/ CLO3		PLO5/ CLO3	PLO5/ CLO3	PLO5/ CLO3
Name	**Reg. No**	**Viva / Quiz / Lab Performance 5 Marks**	**Analysis of data in Lab Report 5 Marks**	**Modern Tool Usage 5 Marks**	**Ethics and Safety 5 Marks**	**Individual and Team Work 5 Marks**
Myesha Khalil	**305093**					
Noor Ansar	**284825**					

Objective:

The objective of open-ended labs is to carry out research, design, and implementation of a problem which covers the all course of Industrial Process Control. That activity is based on following attributes:

1. The activity requires abstract thinking and critical analysis in order to complete the task.
2. The activity requires strong knowledge about the course and also involves creative use of research-based knowledge.
3. The activity covers the individual as well as group involvement to fulfill the task.

To ensure and to make the students capable of applying course knowledge on real world problems, that activity is designed. In that activity you have to design the network or task according to instructor's instructions.
In order to attain the full credit you have to design suitable ladder diagram and download it in your hardware to run in real time in a limited time **(1 Hour)** and then, there will be viva too.

Introduction:

The Internet of Things (IoT) describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. Over the past few years, IoT has become one of the most important technologies of the 21st century. Now that we can connect everyday objects—kitchen appliances, cars, thermostats, baby monitors—to the internet via embedded devices, seamless communication is possible between people, processes, and things.

By means of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyper connected world, digital systems can record, monitor, and adjust each interaction between connected things.

What technologies have made IoT possible?
While the idea of IoT has been in existence for a long time, a collection of recent advances in a number of different technologies has made it practical.

- **Access to low-cost, low-power sensor technology**. Affordable and reliable sensors are making IoT technology possible for more manufacturers.
- **Connectivity.** A host of network protocols for the internet has made it easy to connect sensors to the cloud and to other "things" for efficient data transfer.

- **Cloud computing platforms.** The increase in the availability of cloud platforms enables both businesses and consumers to access the infrastructure they need to scale up without actually having to manage it all.
- **Machine learning and analytics.** With advances in machine learning and analytics, along with access to varied and vast amounts of data stored in the cloud, businesses can gather insights faster and more easily. The emergence of these allied technologies continues to push the boundaries of IoT and the data produced by IoT also feeds these technologies.
- **Conversational artificial intelligence (AI).** Advances in neural networks have brought natural-language processing (NLP) to IoT devices (such as digital personal assistants Alexa, Cortana, and Siri) and made them appealing, affordable, and viable for home use.

What are IoT applications?

IoT Intelligent Applications are prebuilt software-as-a-service (SaaS) applications that can analyse and present captured IoT sensor data to business users via dashboards. We have a full set of IoT Intelligent Applications.
IoT applications use machine learning algorithms to analyse massive amounts of connected sensor data in the cloud. Using real-time IoT dashboards and alerts, you gain visibility into key performance indicators, statistics for mean time between failures, and other information. Machine learning–based algorithms can identify equipment anomalies and send alerts to users and even trigger automated fixes or proactive counter measures.

With cloud-based IoT applications, business users can quickly enhance existing processes for supply chains, customer service, human resources, and financial services. There's no need to recreate entire business processes.

Student Activity:

You are provided a kit containing nodemcu, motion sensors and infrared sensors. You can use Arduino IoT clout service to connect your nodemcu to cloud and be able to control physical GPIO's using android application anywhere from the world.
Your task is to configure nodemcu with any of the available sensors of your choice and configure it to turn on an LED connected to one of the GPIO pins when some specific conditions are met for the sensor used. The same led out should be controllable through mobile application.
Attach your screenshots and codes below and submit report on LMS portal on spot. You can use any resource available online for reference.

(Further explanation in class)

Conclusion:

Task 1:

Code:

```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Untitled 2"
  https://create.arduino.cc/cloud/things/6a8e8d4f-0f71-4018-a483-99ecb96d9950

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the
Thing

  bool lED_switch;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"
const int ledlight=2;
const int sensor=4;
void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  pinMode(sensor,INPUT);
pinMode(ledlight,OUTPUT);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
     The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and errors
     the higher number the more granular information you'll get.
     The default is 0 (only errors).
     Maximum is 4
 */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
```

```
}

void loop() {
  ArduinoCloud.update();
  // Your code here

}

/*
  Since LEDSwitch is READ_WRITE variable, onLEDSwitchChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLEDSwitchChange()  {
 if(lED_switch ==1)
 {digitalWrite(ledlight,HIGH);
   \
 }
 else
 {
   digitalWrite(ledlight,LOW);

 }
}
```

Screenshots:

Task 2 code

```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Untitled 2"
  https://create.arduino.cc/cloud/things/6a8e8d4f-0f71-4018-a483-99ecb96d9950

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the
Thing

  bool lED_switch;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"
const int ledlight=2;
```

```
const int sensor=4;
void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  pinMode(sensor,INPUT);
pinMode(ledlight,OUTPUT);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
     The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and errors
     the higher number the more granular information you'll get.
     The default is 0 (only errors).
     Maximum is 4
 */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  if(digitalRead(sensor)==1){lED_switch=1;
    onLEDSwitchChange();
  }
  else{lED_switch=0;
    onLEDSwitchChange();
  }

}

/*
  Since LEDSwitch is READ_WRITE variable, onLEDSwitchChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLEDSwitchChange()  {
 if(lED_switch ==1)
 {digitalWrite(ledlight,HIGH);
   \
 }
 else
 {
   digitalWrite(ledlight,LOW);
```

```
 }
}
```