

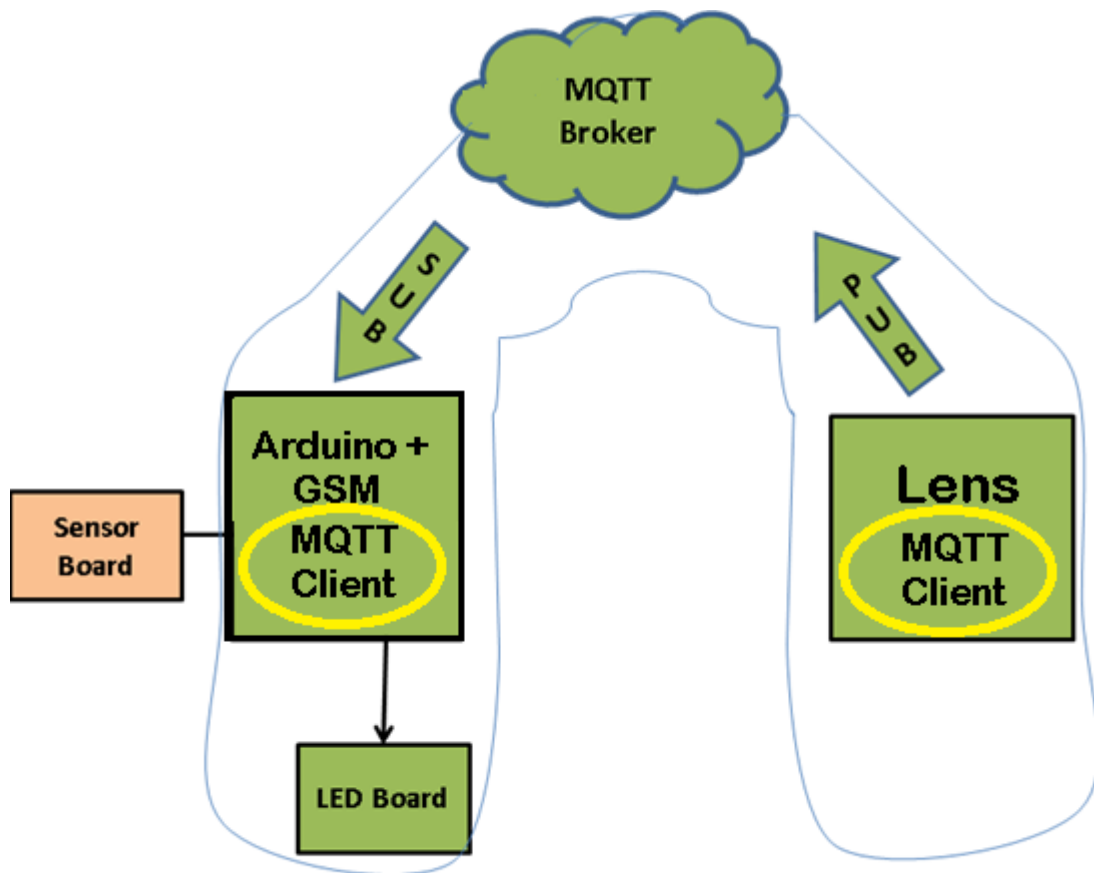
User Manual on

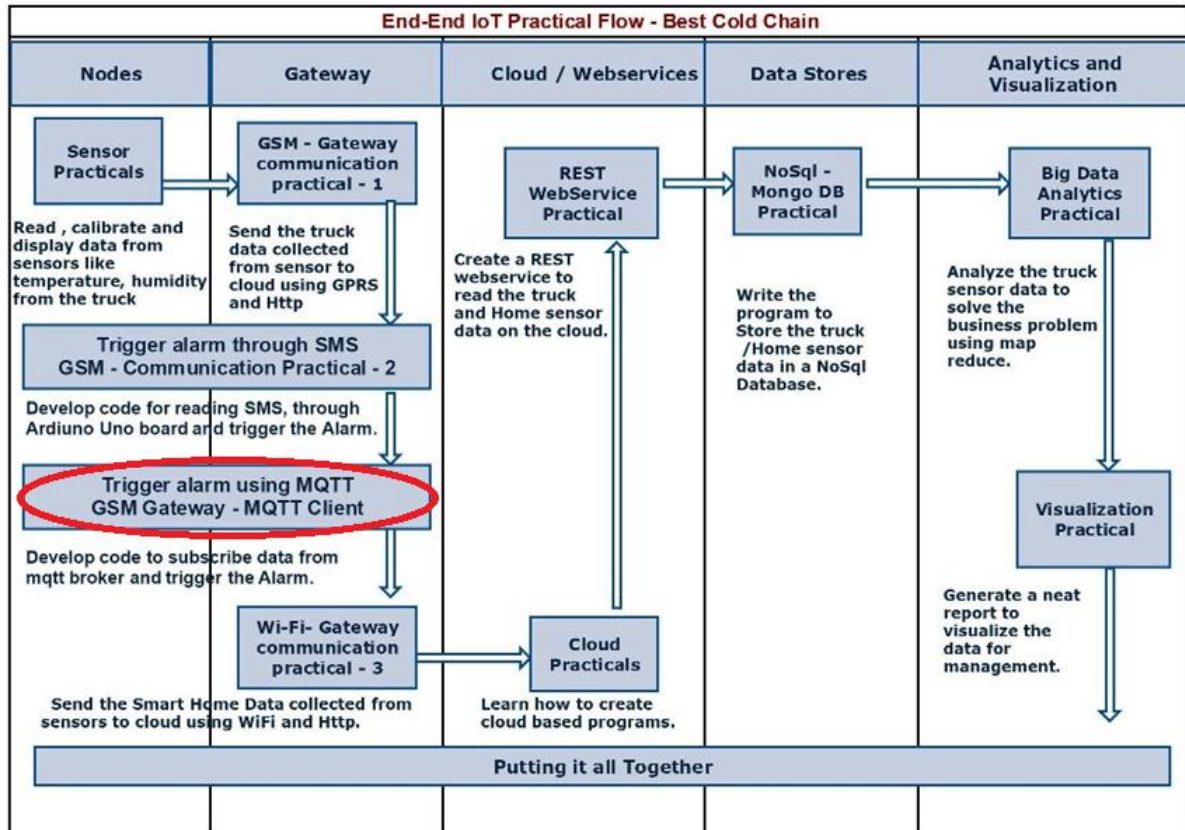
Trigger alarm using MQTT

Practical's Objective:

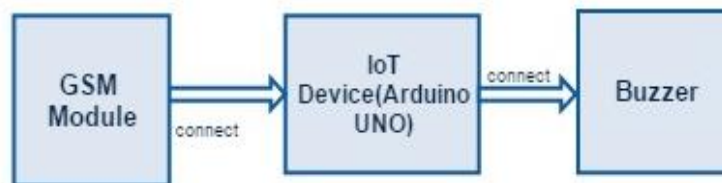
To glow an LED (or trigger an alarm) on Arduino board using an IOT Platform.

1. End-End IoT Flow diagram:

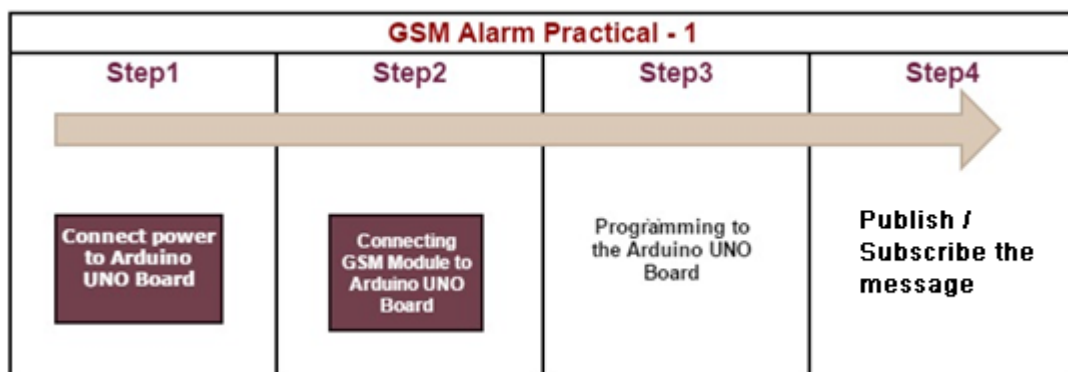




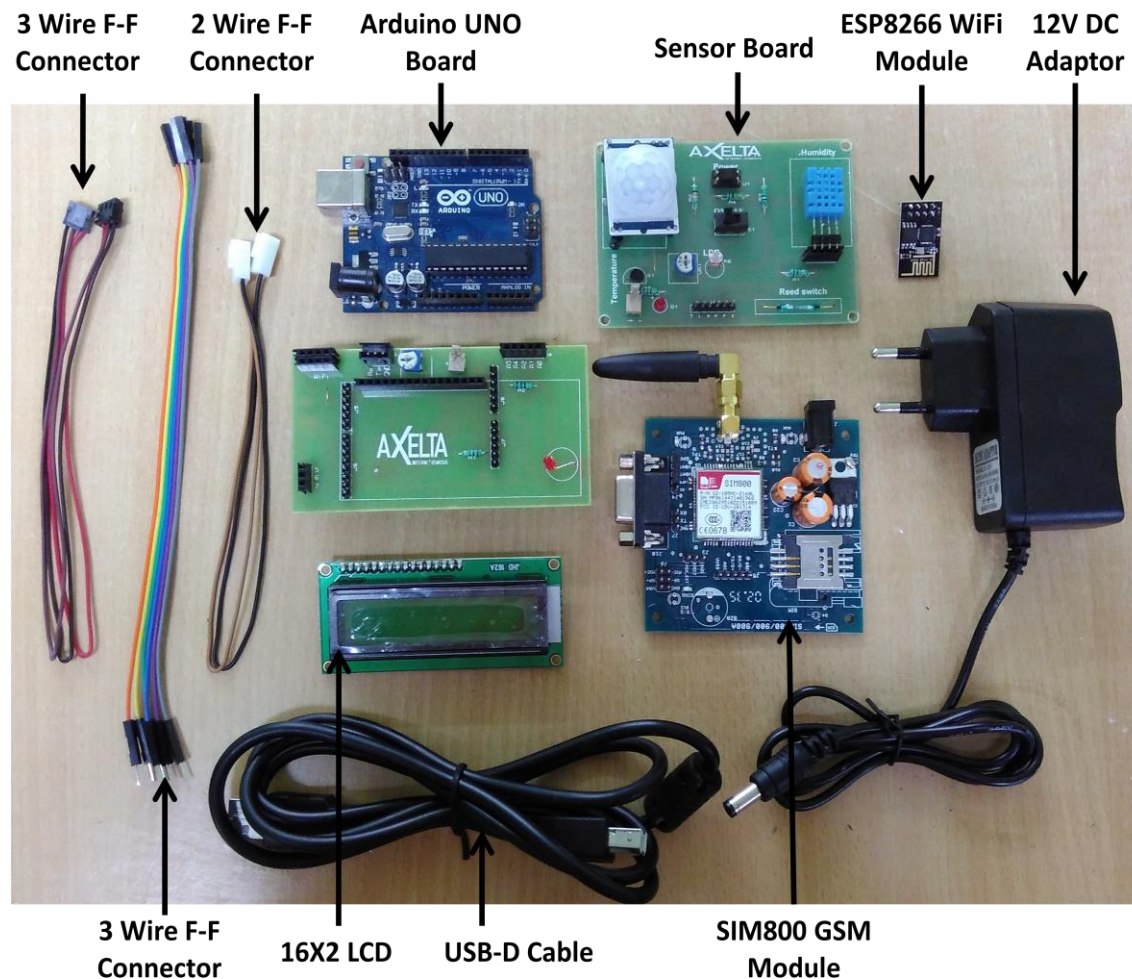
GSM at Node Flow Diagram:



2. GSM MQTT Practical:



3. Hardware requirements:



Software Requirement:

Same as that of previous Practical

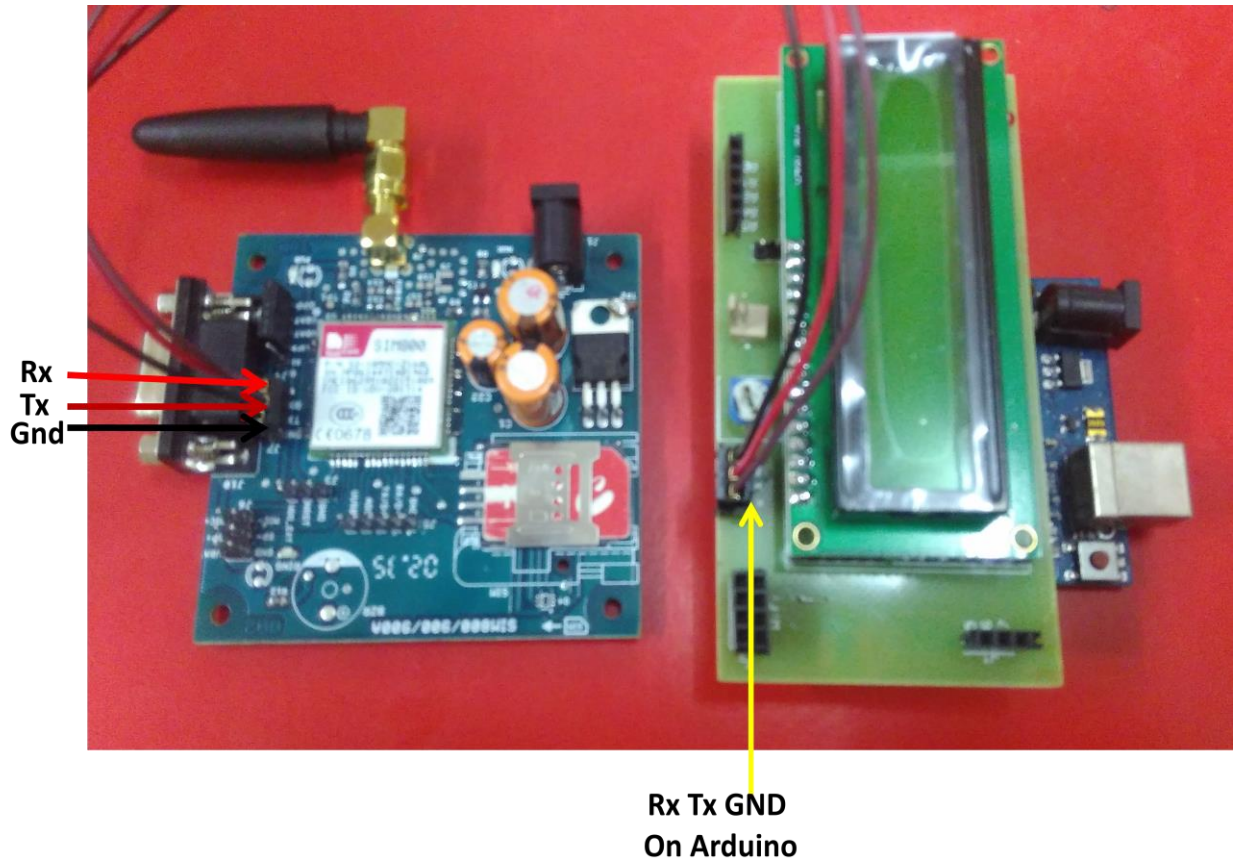
Arduino UNO Board Connections:

Don't change the sensor connections. They will also remain same .

Connecting GSM Modem to Arduino UNO Board

- The Arduino UNO board has a 3-wire connector on the left side.
- Connect this **3-wire connector to the GSM board**. It is a one to one placement connector, which can be connected only in one way.
- Insert a **SIM card in the GSM modem** and make sure it has GPRS activated and sufficient recharge/money in it.

- Connect **12V Adapter to GSM Modem** & power the board then wait for signal (We can find on board signal by checking the Blinking LED on GSM board which shifts its frequency from fast pace to slow pace) .
- Now connect USB cable from Arduino UNO board to PC.



Programming:

Click the link: <https://drive.google.com/file/d/0Bz7GE98wyjOxSUltQ3BMUKjxWTA/view>

If you set **baud rate as 9600 for GSM** You will get the following output on serial monitor :

COM14 (Arduino/Genuino Uno)

Axelta Systems MQTT - LED - ON/OFF

```

AT
OK
+++ATATE1
OK
AT+CREG?
OK
+CREG: 0,1

OK
AT+CIPMUX=0AT+CIPMODE=1
OK
AT+CGATT?
OK
AT+CGATT=1
+CGATT: 1

OK
AT+CIPSTATUS
OK

OK

STATE: IP INITIAL
OAT+CSIT="INTERNET"
OK
AT+CIPSTATUS
OK

STATE: IP START
OAT+CIICR
OK
AT+CIPSTATUS
OK
  
```

1 Response for AT commands

COM14 (Arduino/Genuino Uno)

```

STATE: IP GPRSACT
OAT+CIFSR
100.120.124.150
AT+CIPSTATUS
OK

STATE: IP STATUS
OAT+CIPSTART="TCP","osmosis.axelta.com","1883"
OK

CONNECT
MQTT.TCP_Flag = True
2
2
Connect Acknowledgment
Connection Accepted
3
3
Subscribe Acknowledgment
Message ID :1

0
0
PING Response

0
0
PING Response

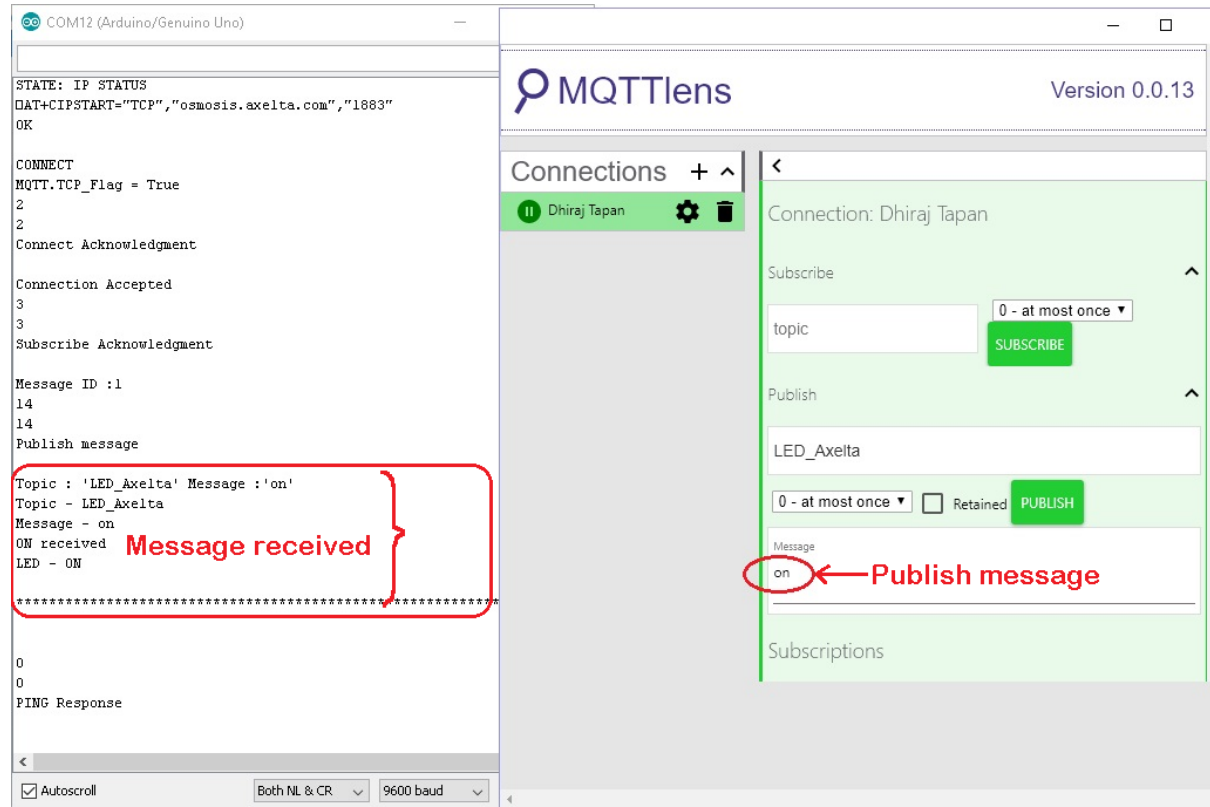
0
0
PING Response

0
0
PING Response
  
```

2

3 Connected to MQTT Broker

4 subscription done



1. Publisher (**here MQTT client**) will publish a message on the topic (**LED_Axelta in our Program**) to MQTT Broker.
2. Broker will send the same message to all the **MQTT Clients (here Arduino with GSM Board)** who have subscribed for the same topic.
3. Based on the message received the Arduino will take the following action :

Message	Action taken by Arduino
on	LED will turn on
off	LED will turn off
blink	LED will blink for 2 seconds then will turn off again

Step by Step Explanation for the Program

```
//***** Include all the necessary Header Files*****//
```

```
#include <SoftwareSerial.h>
```

```
#include <GSM_MQTT_Axelta.h>
```

```
#include <LiquidCrystal.h>
```

```
//*****Make the global declaration of the UART pins with the microcontroller pins of  
the Arduino board, i.e., RX, TX with pins 3, 4 using "SoftwareSerial". *****//
```

```
SoftwareSerial Axelta(3, 4);
```

```
//*****Make the global declaration of the LCD pins with the microcontroller pins of the  
Arduino board, i.e., RS, E, D4, D5, D6, D7, with pins 8, 9, 10, 11, 12, 13 using  
"LiquidCrystal" *****//
```

```
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
```

```
// LED Connected to GPIO pin A3 of Arduino
```

```
int LEDpin = A3;
```

```
//***** Write correct APN as per to your country *****//
```

```
char* APN = "INTERNET";
```

```
String responseON = "on";
```

```
String responseOFF = "off";
```

```
String responseBLINK = "blink";
```

```
String MQTT_HOST = "osmosis.axelta.com";
```

```
String MQTT_PORT = "1883";
```

```
char* ClientId = "Axelta-MQTT-Dhiraj";
```

```
char* SubTopic = "LED_Axelta";
```

```
//*****//
```

```
GSM_MQTT_Axelta MQTT(120);
```

```
void GSM_MQTT_Axelta::AutoConnect(void)
{
    connect(ClientId, 0, 0, "", "", 1, 0, 0, 0, "", ""); // void connect(char *ClientIdentifier, char
UserNameFlag, char PasswordFlag, char *UserName, char *Password, char CleanSession,
char WillFlag, char WillQoS, char WillRetain, char *WillTopic, char *WillMessage);
}
```

```
void GSM_MQTT_Axelta::OnConnect(void)
{
    subscribe(0, _generateMessageID(), SubTopic, 1); // void subscribe(char DUP, unsigned
int MessageID, char *SubTopic, char SubQoS);
}
```

//***Write a Boolean function as “find_string” with String parameters as “base, search” The function “find_string” will check if the received data is equal to “on” *******
*******//**

```
boolean find_string(String base, String search)
{
    int len = search.length(); // find the length of the base string
    for(int m = 0; m<((base.length()-len)+1);m++)// Iterate from the beginning of the base string
till the end minus length of the substring
    {
        if(base.substring(m,(m+len))==search) // Check if the extracted Substring Matches the
Search String
        {
            return true; // if it matches exit the function with a true value
        }
    }
    return false; // if the above loop did not find any matches, control would come here and
return a false value
}
```

//*** Define a function GSM_MQTT_Axelta::OnMessage as soon as any message received on subscribed topic *****//**

```
void GSM_MQTT_Axelta::OnMessage(char *Topic, int TopicLength, char *Message,
```



```
int MessageLength)
{
  Serial.print("Topic - ");
  Serial.println(Topic);
  Serial.print("Message - ");
  Serial.println(Message);

  if(find_string(Message,responseON))
  {
    digitalWrite(LEDpin, HIGH);
    Serial.println("ON received");
    Serial.println("LED - ON");
    lcd.clear();
    lcd.print("ON received");
    lcd.setCursor(0, 1);
    lcd.print("LED ON ");
    delay(300);
    Serial.println();

    Serial.println("*****");
    Serial.println();
    Serial.println();
  }

  else if(find_string(Message,responseOFF))
  {
    digitalWrite(LEDpin, LOW);
    Serial.println("OFF received");
    Serial.println("LED - OFF");
    lcd.clear();
    lcd.print("OFF received");
    lcd.setCursor(0, 1);
    lcd.print("LED OFF ");
    delay(300);
    Serial.println();

    Serial.println("*****");
    Serial.println();
  }
}
```

```
Serial.println();
}

else if(find_string(Message,responseBLINK))
{
digitalWrite(LEDpin, LOW);
Serial.println("BLINK received");
Serial.println("LED - Blink");
lcd.clear();
lcd.print("BLINK received");
lcd.setCursor(0, 1);
lcd.print("LED BLINK");
delay(10);
for(int i=0; i<10; i++)
{
digitalWrite(LEDpin, HIGH);
delay(20);
digitalWrite(LEDpin, LOW);
delay(100);

}
Serial.println();

Serial.println("*****");
Serial.println();
Serial.println();
}

else
{
Serial.println("Command NOT Supported");
lcd.clear();
lcd.print("Command NOT Supported");
delay(300);
Serial.println();

Serial.println("*****");
```

```
Serial.println();  
Serial.println();  
}  
}
```

//***Write a “setup” function and initialize the UART communication with 9600 baud rate and the LCD as 16 columns with 2 rows via the “lcd.begin” function.*****//**

```
void setup()  
{  
  Serial.begin(9600);  
  Axelta.begin(9600);  
  delay(3000);  
  lcd.begin(16, 2); // Begin LCD 16 Columns and 2Rows  
  pinMode(LEDpin, OUTPUT);  
  digitalWrite(LEDpin, LOW);  
  Serial.println("Axelta Systems MQTT - LED - ON/OFF");  
  Serial.println();  
  lcd.clear();  
  lcd.print("Axelta Systems MQTT - LED - ON/OFF");  
  delay(1000);  
  MQTT.begin();  
}
```

//***Start a function with the name “loop”. This is the Main loop/function of the whole code/program****//**

```
void loop()  
{  
  if (MQTT.available())  
  {}  
  MQTT.processing();  
}
```
