

Cloudy Weather Phone Call Alert System

Materials Required:

1. Node MCU ESP8266(has Wi-Fi module)
2. Arduino Uno R3(For reading sensor value and sending it to Node MCU)
3. LDR Sensor (For measuring the light intensity)
4. Connecting Wires
5. Arduino IDE
6. IFTTT App (For making phone call to your number)

Problem Statements:

1. Let us suppose that you work in a firm that is far away and you live alone at your home. You have kept your clothes to dry in the sunlight and went to office. Now if weather gets cloudy at your home, then you would wish to get informed so that you can call some of your neighbours to keep the cloth inside their house.
2. Let us take another situation in which you have a device (say X) that recharges using solar energy. You have kept it outside in the sun to charge and went to do other work. Suppose the weather gets cloudy and due to some other work, you forgot that you have kept X for charging outside in the open. Now if it rains then device X might get internally damaged. So, to stop this you wish that your device should call you and speak "The weather

is cloudy near <material* name>. Keep it inside” as soon as the weather gets cloudy.

Solution Algorithm (a.k.a. How it works):

1. Keep LDR sensor at the place where your clothes or device X is placed.
2. Read the light intensity value from LDR sensor into Arduino A0 pin and send it to Node MCU.
3. As soon as weather becomes cloudy the Arduino would read less voltage in A0 as the light intensity has decreased.
4. When the light intensity falls below a certain threshold. Node MCU receives this value and sends a request to IFTTT server to trigger an applet (shown in fig. 2) there.
5. I have built an applet on IFTTT such that if it receives a request it will call me and say “The weather is cloudy near <material name>. Keep me inside”. It will call twice such that if the user misses one call he can receive the other.

Requirement of Wi-Fi: For Node MCU to send request to IFTTT server after the output voltage of the LDR sensor falls below a certain level, we need Internet. So, to connect our Node MCU to internet we require a Wi-Fi module. That Wi-Fi module is ESP8266. Although if one is at home then certainly Wi-Fi is not so necessary. He/she can use a buzzer instead sending alert call to his

phone. But I have designed this project keeping in mind that if people are at home then most of the time they can see that the weather has become cloudy, but if they have gone far away (more than 20-40 km from their home) then the weather at their home and at the place where they are at present can be very different (I have myself experienced this). So, they definitely need wireless alert system (I have used alert call here), and wired alert system using buzzer won't work (Why? **). **Hence, the mentioned system has real requirement of Wi-Fi.**

Circuit Diagram:

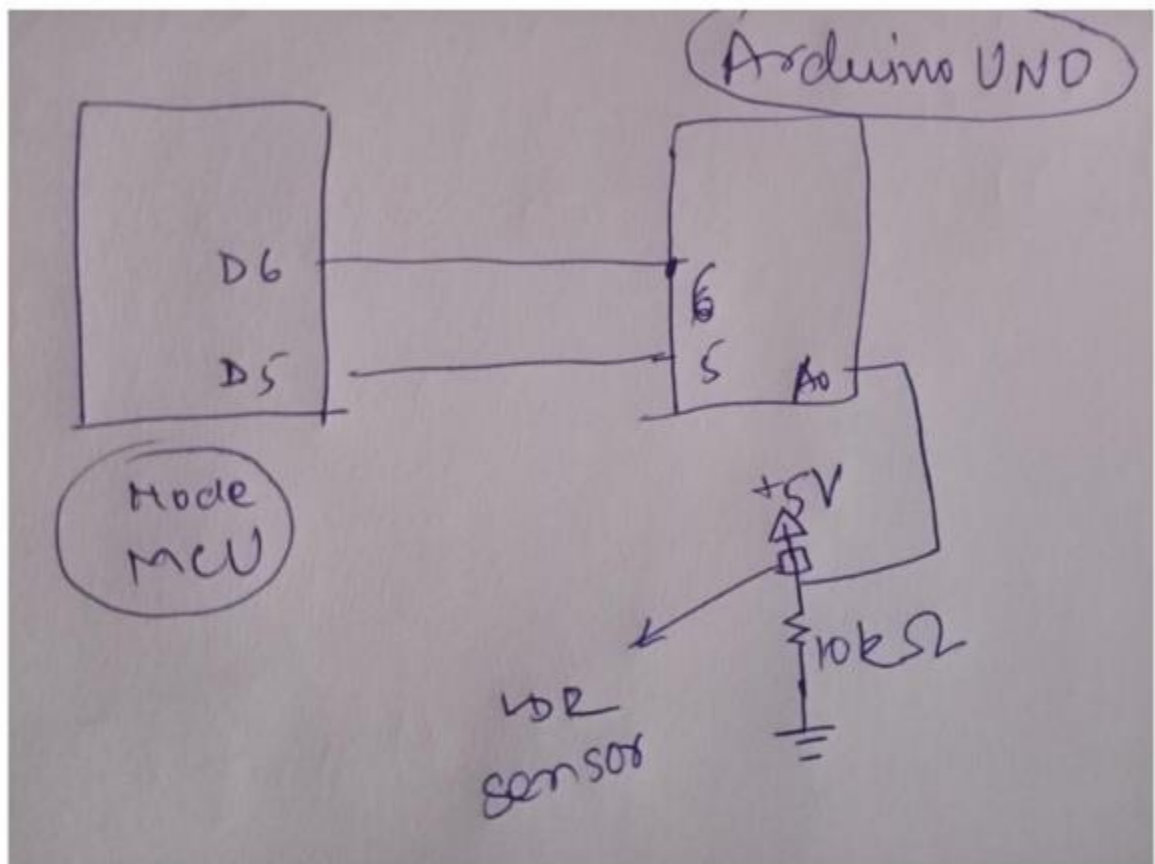


Fig 1(circuit diagram)

- The digital pin 6 and 5 of the Arduino Uno and D5 and D6 pins of the Node MCU are connected to each other as shown in the figure. This connection is made so that the value read at A0 is communicated to the Node MCU.
- One terminal of the LDR sensor is connected to the 5 volt of the Arduino and the other end is connected to ground of the Arduino using a 10kilo ohm pull down resistor
- The point between 10k and the LDR sensor is connected to the A0 analog pin of the Arduino (the voltage read is proportional to the intensity of light falling on it).

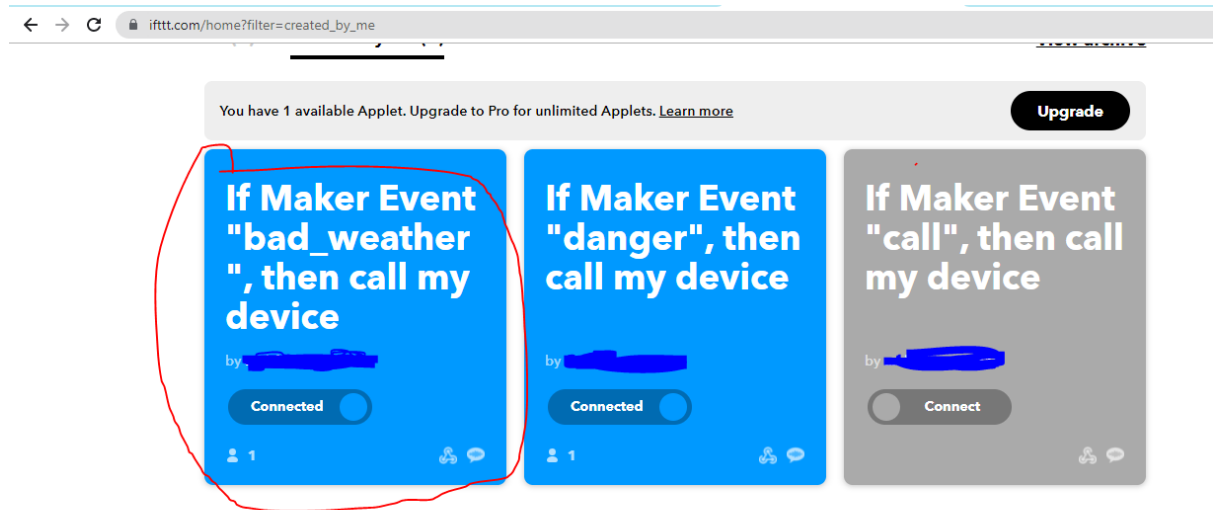


Fig 2(my applets on IFTTT, first circled applet is used here)

Read the comments* to understand the working of each line of this code:**

Arduino Code:

```
#include <SoftwareSerial.h>

SoftwareSerial uno(5,6); //5-receiver 6-transmitter

void setup() {
  uno.begin(4800); // begins serial communication
}

void loop() {
  int data=analogRead(A0);

  /*If the arduino receives 's' then it sends the value read at analog pin
  A0 to the node mcu*/
  if(uno.available()>0)
  {
    char c=uno.read();
    if(c=='s'){
      uno.write(data);
    }
  }
}
```

Node MCU Code:

```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>

/* D6- rx and D5 tx. These pins are used for communication between
node mcu and Arduino*/
SoftwareSerial uno(D6,D5);
int light_int;// stores the output voltage from the LDR sensor

/*ssid and password of the network to which node mcu will connect
for internet*/
const char* ssid = "<Your ssid>";
const char* password = "<Your password>";

/* If it's first time you will receive alert call then ring_number value
will be 1. After sending first alert call it's value becomes 2. It is used
to make sure that not more than two alert calls are sent, that is,
when the ring_number becomes 3 request won't be sent to IFTTT
and hence the applet won't be triggered and you won't receive call
after the second call.
*/
int ring_number=1;

/* samay gives the functionality of delay to be set dynamically at the
end of loop function.
```

- * If the first call is triggered
- * then there is more delay as compared to when there is no call triggered. If it was not
- * so then the two calls would be triggered
- * with very less time gap between them. If this happens then as soon as the user picks
- * the first call he would receive the second call.
- * In this way user has to reject the second call to hear the message of the first call.
- * And effectively we would end with only one call.
- * So to prevent this samay variable is used.

*/

```
int samay=500;
```

```
const char* host = "maker.ifttt.com";
```

```
void setup()
```

```
{
```

```
/* initilizes node mcu for serial communication at 48000 baud rate.
```

```
Arduino also need to have the same baud rate.*/*
```

```
uno.begin(4800);
```

```
Serial.begin(9600); //Initilizes the serial monitor for serial  
communication from node mcu
```

```
//Below two lines are for indicating to which network we are  
connecting to
```

```
Serial.println("Connecting to ");
```

```
Serial.println(ssid);
```

```
WiFi.begin(ssid,password);//attempts to connect to wifi with  
internet
```

```
//Below while loop waits till the node mcu is not connected to  
internet
```

```
while(WiFi.status() !=WL_CONNECTED)
```

```
{
```

```
Serial.print(". ");
```

```
delay(1000);
```

```
}
```

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

```
Serial.println(WiFi.localIP());//prints the ip address of node mcu
```

```
}
```



```
void loop()
{

    uno.write("s");// triggers arduino to read and sent the data at A0

    /* below 4 lines checks whether there is response from the arduino,
    * if yes then node mcu reads and stores the response in ligh_int
    variable
    * further it also prints un the serial monitor for deveoloper's
    reference
    */
    if (uno.available()>0)
    {
        light_int=uno.read();
        Serial.println(light_int);
    }

    WiFiClient client; // Our node mcu is acting as client

    const int httpPort = 80; // this port is used for communication
    through web browsers
```

```

    if (!client.connect(host, httpPort)) // attempts to connect to
server till the connection is successful

    {
        Serial.println("connection failed");
        return;
    }

    // request is sent to the server which triggers the applet on IFTTT
to make the user an alert call

    if(light_int<200 && ring_number<3)

    {
        ring_number=ring_number+1;
        String url =
"/trigger/danger/with/key/lvFa63YcGh0We3HMqhdBCxovzql2lATqC
4KrsURrN2q";
        Serial.print("Requesting URL: ");
        Serial.println(url);
        client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " +
host + "\r\n" + "Connection: close\r\n\r\n");
        samay=20000;
    }

    delay(samay);
}

```

* Here material can be clothes that are kept in the sunlight for drying or some solar rechargeable light

** Just imagine how impractical it is to carry wires with you all around.

*** Single line comments in the code are written after `//` and multiline line comments are written between `/*` and `*/`. The comments are in blue color