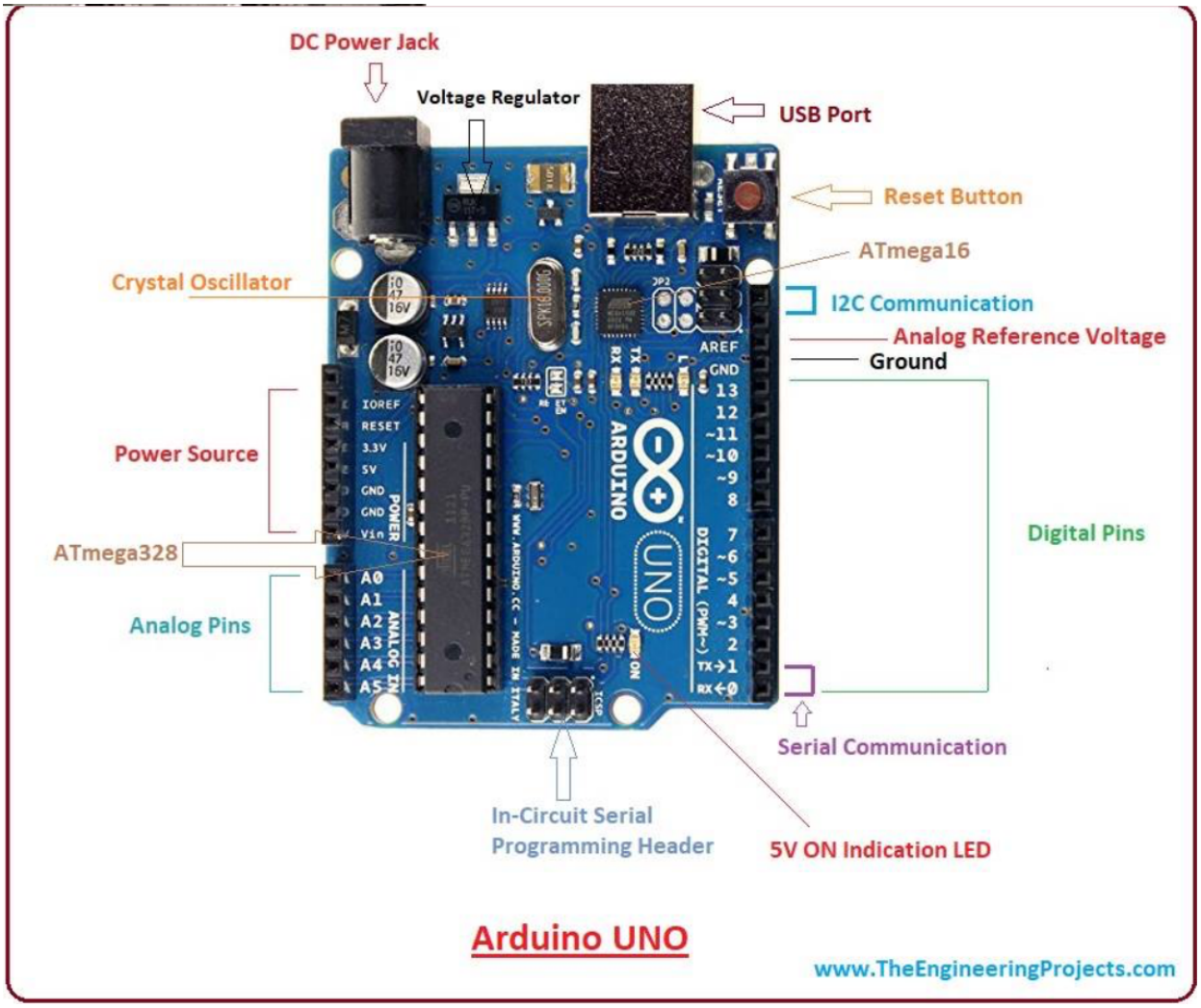
Arduino IOT Report

Rishit Gupta

What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.



Given above is the pin diagram of the Arduino UNO.

Parameters	Arduino Uno
GPIO Pins	14 Pins(0-13)
Analog Pins	6 Pins (A0-A5)
Tx Rx Pins	0(Rx), 1(Tx)
RAM	32K bytes
ROM	1KB(EEPROM)
Peripherals	
PWM pins	5 out of the 14 GPIO pins
Frequency(Clock Speed)	16MHz
SRAM	2KB
ADC(bits,channels)	10 bit, 6 channels

Various parameters of the Arduino Uno

Arduino Programmes and Interfacings

1) Blink

This code turns an LED on and off after every 1 second.

```
void setup(){
pinMode(0,OUTPUT);
pinMode(1,OUTPUT);
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
                }

void loop(){
digitalWrite(0,HIGH); digitalWrite(1,HIGH); digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
delay(1000);
digitalWrite(0,HIGH); digitalWrite(1,HIGH); digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
delay(1000);
                }
```

2) Multi Blink using Arrays

The following code turns on and off the different pins of the LED's one by one depending upon how they are placed in the array.

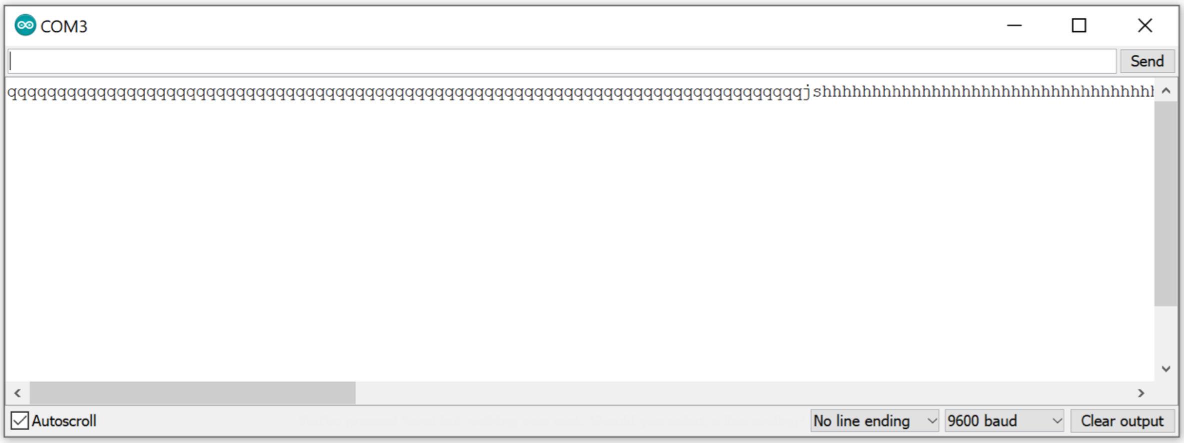
```
int timer = 1500;          // The higher the number, the slower the timing.
int ledPins[] = {
  2,1,0,6,5,3};
int pinCount = 6;          // the number of pins (i.e. the length of the array)
void setup() {
for (int thisPin = 0; thisPin < pinCount; thisPin++) {
pinMode(ledPins[thisPin], OUTPUT);
                                                        }}
void loop() {
for (int thisPin = 0; thisPin < pinCount; thisPin++) {
digitalWrite(ledPins[thisPin], HIGH);
delay(timer);
digitalWrite(ledPins[thisPin], LOW);                    }
for (int thisPin = pinCount - 1; thisPin >= 0; thisPin--) {
digitalWrite(ledPins[thisPin], HIGH);
delay(timer);
digitalWrite(ledPins[thisPin], LOW);
                                                    }
                }
```

3) Serial Read and Write

a) Serial Write

```
int a=0;
void setup(){
Serial.begin(9600);
            }
void loop(){
Serial.print("Hiiiii Helooo ");
Serial.print(" Hahahaha");
Serial.println(a);
a=a+1;
delay(1000);
            }
```

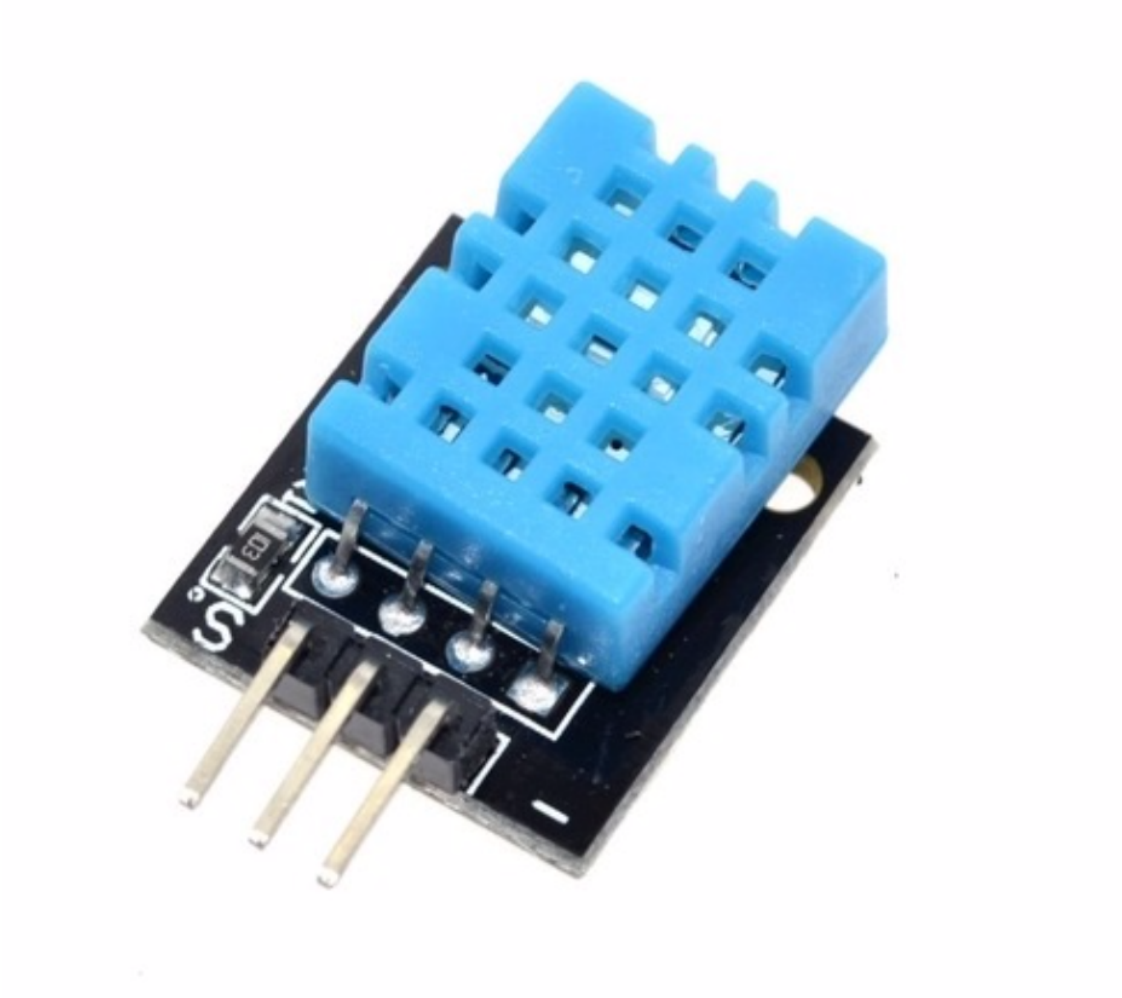
```
void setup() {
  Serial.begin(9600);
}

void loop (void) {
  if (Serial.available() > 0) Serial.print((char)Serial.read());
}
```



b) Serial Read

```
int inByte=0;
void setup(){
Serial.begin(9600);
            }
void loop(){
if(Serial.available(>0)){
inByte=Serial.read();
Serial.write(inByte);
delay(20);
            }
}
```

3) Temperature and Humidity Sensor DHT11



DHT11 is a sensor used to measure the temperature and humidity.

```
#include "DHT.h"
#define DHTPIN 7     // what pin we're connected to
#define DHTTYPE DHT11   // for DHT 22 only
DHT dht(DHTPIN, DHTTYPE,3);
float voltage0=0;
float temp=0;
float voltage5=0;
float light=0;
void setup() {
 Serial.begin(9600);
Serial.println("DHT22 test  out ==");
dht.begin();
                }
void loop() {
delay(2000);
```

```
float hi = dht.readHumidity();
float temp = dht.readTemperature();
 if (isnan(hi) || isnan(temp) ) {
Serial.println("Failed to read from DHT sensor! check sensor , clock
speed , connections");
return;
                                        }
int sensorValue5 = analogRead(A5);
voltage5=sensorValue5*4.887;
light=voltage5/5000;
light=light*100;
Serial.print("Light = ");
Serial.print(light);
Serial.println(" %");
Serial.println("---------------------------------");
Serial.print("Humidity: ");
Serial.print(hi);T
Serial.println(" %");
Serial.print("Temperature: ");
Serial.print(temp);
Serial.println(" *C ");
Serial.println("---------------------------------");
Serial.println("---------------------------------");
                        }
```

4) Temperature Sensor (ds18b20)


```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 7
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float voltage5=0;
float light=0;
void setup(void)
{
 Serial.begin(9600);
 Serial.println(" Temperature sensor ds18b20");
 sensors.begin();
}
void loop(void)
{
int sensorValue5 = analogRead(A5);
voltage5=sensorValue5*4.887;
light=voltage5/5000;
light=light*100;
Serial.print("Light = ");
Serial.print(light);
Serial.print("Requesting temperatures...");
sensors.requestTemperatures(); // Send the command to get temp.
Serial.println("DONE");
float a= sensors.getTempCByIndex(0);
// After we got the temperatures, we can print them here.
```

```
// We use the function ByIndex, and as an example get the temperature
from the first sensor only.
Serial.print("Temperature for the device 1 (index 0) is: ");
Serial.println(a);
delay(2000);
}
```

5) PWM LED Fade

This code controls the intensity of light in an LED. The intensity is set between 0-255.

```
void setup(){
pinMode(3,OUTPUT);
pinMode(5,OUTPUT);
analogWrite(3,128); //pwm 8 bit 0-255
delay(3000);
                }
void loop(){
analogWrite(3,250); //pwm 8 bit 0-255
delay(3000);
analogWrite(5,255);
delay(3000);
analogWrite(3,90); //pwm 8 bit 0-255
delay(3000);
analogWrite(5,128);
delay(3000);
                }
```

6) PWM LED Fade continuous fade

This code controls the intensity of light dynamically in an LED. It first dims the LED and then it increases the brightness.

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 10;    // how many points to fade the LED by

void setup() {
pinMode(led, OUTPUT);
pinMode(3,OUTPUT);
                }
void loop() {
analogWrite(led, brightness);
analogWrite(3, brightness);
brightness = brightness + fadeAmount;
if (brightness <= 0 || brightness >= 255) {
fadeAmount = -fadeAmount;
                                                }
delay(100);
                }
```

7) Analog Read

This is a basic code of reading the analog value and converting into millivolts.

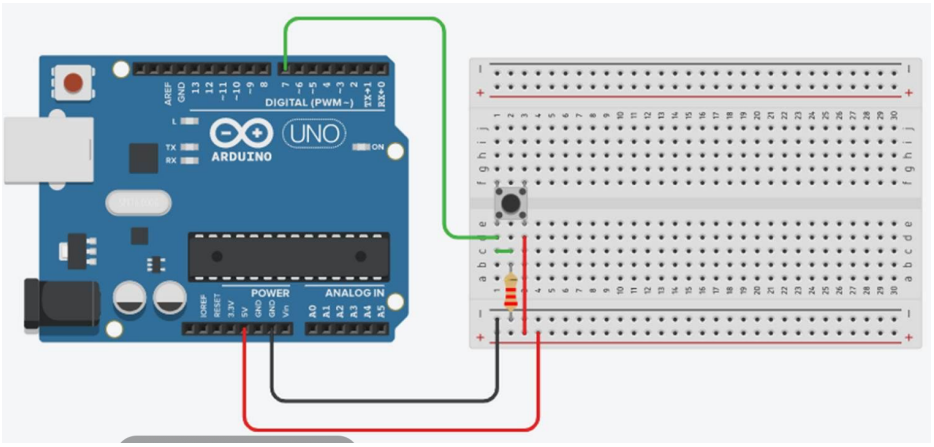
```
float voltage0=0;
float voltage1=0;
float voltage2=0;

void setup() {
 // initialise serial communication at 9600 bits per second:
Serial.begin(9600);
Serial.println("reading raw value at analog pin A0,A1,A2");
delay(5000);
                }

void loop()
{
int sensorValue0 = analogRead(A0);
int sensorValue1 = analogRead(A1);
int sensorValue2 = analogRead(A2);
voltage0=sensorValue0*4.887;
voltage1=sensorValue1*4.887;
voltage2=sensorValue2*4.887;
Serial.println(sensorValue0);
Serial.print("the value of voltage on A0 pin ==");
Serial.print(voltage0);
Serial.println(" mV");
Serial.println(sensorValue1);
Serial.print("the value of voltage on A1 pin ==");
Serial.print(voltage1);
Serial.println(" mV");
Serial.println(sensorValue2);
Serial.print("the value of voltage on A2 pin ==");
Serial.print(voltage2);
Serial.println(" mV");
delay(1000);
}
```

8) Button Press

In this code whenever the button is pressed, the LED is turned on. Whenever it isn't pressed, the LED remains off.

```
int buttonPin=2;
int ledPin1=5;
int ledPin2=6;
int ledPin3=7;
int buttonState=0;
void setup(){
pinMode(ledPin1,OUTPUT);
pinMode(ledPin2,OUTPUT);
pinMode(ledPin3,OUTPUT);
pinMode(buttonPin,INPUT);
                }

void loop(){
buttonState=digitalRead(buttonPin);
if(buttonState==1){
        digitalWrite(ledPin1,1);
        digitalWrite(ledPin2,0);digitalWrite(ledPin3,1);
        delay(1000);
        digitalWrite(ledPin1,0);
        digitalWrite(ledPin2,1);digitalWrite(ledPin3,0);
        delay(1000);
}
else{
        digitalWrite(ledPin1,0);
        digitalWrite(ledPin2,0);digitalWrite(ledPin3,0);
        delay(1000);
        digitalWrite(ledPin1,1);
        digitalWrite(ledPin2,1);digitalWrite(ledPin3,1);
        delay(1000);
       }
                }
```

9) Button Press with PWM pins

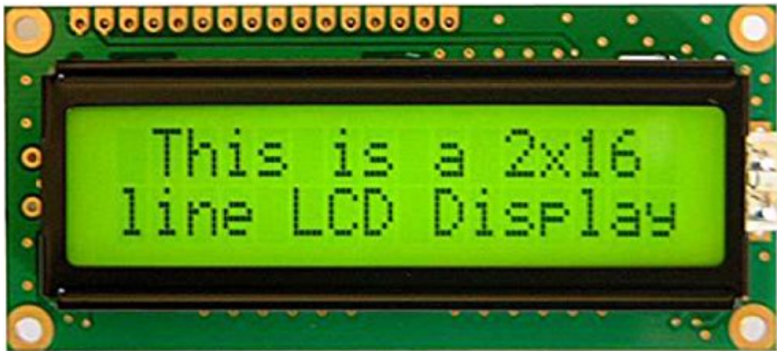


In this code whenever the button is pressed, the LED is brightened. Whenever it isn't pressed, the LED becomes dim.

```
int buttonPin=2;
int ledPin1=5;
int ledPin2=6;

int buttonState=0;

void setup(){
pinMode(ledPin1,OUTPUT);
pinMode(ledPin2,OUTPUT);
pinMode(buttonPin,INPUT);
                }

void loop(){
buttonState=digitalRead(buttonPin);
if(buttonState==0){
        analogWrite(ledPin1,200);
        analogWrite(ledPin2,200);
                        }
else{
        analogWrite(ledPin1,50);
        analogWrite(ledPin2,50);
}
}
```

10) LCD

In this code we have interfaced a 16x2 LCD display and displayed the millivolts on the screen.




```
#include <Wire.h>   // i2c start working
#include <LiquidCrystal_I2C.h> // i2c lcd lib
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
 // initialize the LCD
 lcd.begin();
 // Turn on the blacklight and print a message.
 // lcd.backlight();
lcd.setCursor(0,0);
lcd.setCursor(0,0);
lcd.print("Display");
lcd.setCursor(0,1);
lcd.print("analog value");
delay(4000);
lcd.clear();
}
void loop()
{
int sensorValue1 = analogRead(A1);
int sensorValue0 = analogRead(A0);
float voltage1=sensorValue1*4.887;
float voltage0=sensorValue0*4.887;
lcd.setCursor(0,0);
lcd.print("Voltage0: ")
lcd.setCursor(10,0);
lcd.print(voltage0);
lcd.setCursor(0,1);
lcd.print("Voltage1: ")
```

```
lcd.setCursor(10,1);
lcd.print(voltage1);
delay(2000);
}
```

11) Sending data to the Adafruit cloud


```
#define apin A0
#define IO_USERNAME    "rishit_31"
#define IO_KEY         "aio_qtJm21GoZyxBkEAjh9bfzeROHJyn"
#define WIFI_SSID       "NetgearHome"
#define WIFI_PASS       "rishit2000"
#include "AdafruitIO_WiFi.h"
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
int current = 0;
int last = -1;
AdafruitIO_Feed *analog = io.feed("Rishit_69");
void setup() {
Serial.begin(115200);
while(! Serial);
Serial.print("Connecting to Adafruit IO");
io.connect();
while(io.status() < AIO_CONNECTED) {
Serial.print("-");
delay(500);                                          }
Serial.println();
Serial.println(io.statusText());
                                                   }
void loop() {
io.run();
current = analogRead(apin);  // 0-1023
if(current == last)
```

```
return;
Serial.print("sending  data -> ");
Serial.println(current);
analog->save(current);
last = current;
delay(5000)}
```

12) Sending Temperature, Humidity, Analog Read, to the cloud

```
#define IO_USERNAME    "rishit_31"
#define IO_KEY         "aio_qtJm21GoZyxBkEAjh9bfzeROHJyn"
#define WIFI_SSID       "mywifi"
#define WIFI_PASS       "123456789abc"
#include "AdafruitIO_WiFi.h"
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
#define pin1 D1
#define pin2 D2
#define pin3 D3
AdafruitIO_Feed *digital1 = io.feed("digital1");
AdafruitIO_Feed *digital2 = io.feed("digital2");
AdafruitIO_Feed *digital3 = io.feed("digital3");
#include "DHT.h"
#define DHTPIN D5
#define apin A0
#define DHTTYPE DHT11   // for DHT 22 only
DHT dht(DHTPIN, DHTTYPE,20);
int currentT = 0;
int lastT = -1;
int currentH = 0;
int lastH = -1;
AdafruitIO_Feed *humidity = io.feed("Humidity");
AdafruitIO_Feed *temp = io.feed("Temperature");
void setup() {
pinMode(pin1, OUTPUT);
pinMode(pin2, OUTPUT);
pinMode(pin3, OUTPUT);
Serial.begin(115200);
dht.begin();
while(! Serial);
Serial.print("Connecting to Adafruit IO");
io.connect();
digital1->onMessage(handleMessage1); //
digital2->onMessage(handleMessage2);
digital3->onMessage(handleMessage3);
// wait for a connection
```

```
while(io.status() < AIO_CONNECTED) {
Serial.print("_");
delay(500);
                                                    }

// we are connected
Serial.println();
Serial.println(io.statusText());
digital1->get();
digital2->get();
digital3->get();
                                                     }

void loop() {
 io.run();
 float h = dht.readHumidity();
 // Read temperature as Celsius
 float t = dht.readTemperature();


current = analogRead(apin)*3.225;

if (isnan(h) || isnan(t) ) {
Serial.println("Failed to read from DHT sensor!");
return;
                                  }

if(currentT!=lastT){
Serial.print("sending  data -> ");
Serial.println(currentT);
temp->save(t);
lastT=currentT;
                          }
if(currentH!=lastH){
Serial.print("sending  data -> ");
Serial.println(currentH);
humidity->save(t);
lastH=currentH;
                          }
delay(5000);
 }
void handleMessage1(AdafruitIO_Data *data) {
Serial.print("received data on pin1 <- ");
if(data->toPinLevel() == HIGH)
Serial.println("pin1 high");
```

```
else
Serial.println("pin1 low");
digitalWrite(pin1, data->toPinLevel());// 1 , 0
}
void handleMessage2(AdafruitIO_Data *data) {
Serial.print("received data on pin2 <- ");

if(data->toPinLevel() == HIGH)
Serial.println("pin2 high");
else
Serial.println("pin2 low");
digitalWrite(pin2, data->toPinLevel());// 1 , 0
}
void handleMessage3(AdafruitIO_Data *data) {
Serial.print("received data on pin3 <- ");
if(data->toPinLevel() == HIGH)
Serial.println("pin3 high");
else
Serial.println("pin3 low");
digitalWrite(pin3, data->toPinLevel());// 1 , 0
}
```

12) IFTTT Node MCU Adafruit Google Assistant