PRECISION IN THE AIR: EXPLORING THE DHT11 SENSOR

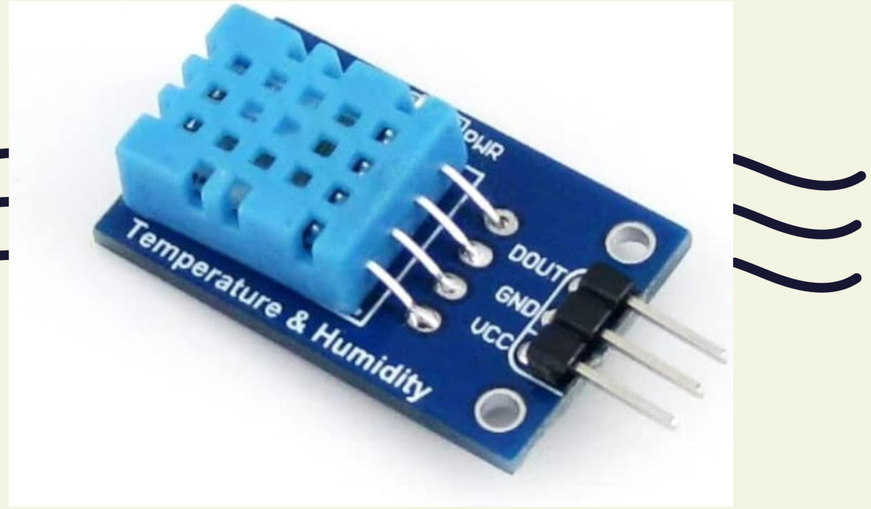
Let's check the temperature and humidity in the room.

TODAY'S FOCUS

INTRODUTCTION TO DHT11

INTERGRATING DHT11 TO ARDUINO IOT CLOUD

DHT FAMILY



The DHT series of sensors, including the DHT11, are widely used for measuring temperature and humidity in various applications. These sensors are cost-effective, easy to use, and provide digital outputs. The DHT11 specifically measures temperature in the range of 0°C to 50°C and humidity in the range of 20% to 90% with moderate accuracy.

The DHT (Digital Humidity and Temperature) family of sensors includes several popular models, such as the DHT11, DHT21 (AM2301), and DHT22 (AM2302). These sensors are widely used for measuring temperature and humidity in a range of applications, from home automation to environmental monitoring.

READING DATA FROM DHT11

Working with the DHT11 sensor requires a library from Adafruit.

The library makes our works very simple and Makes the sensor easy to handle

Let’s See How basic code of DHT11 works

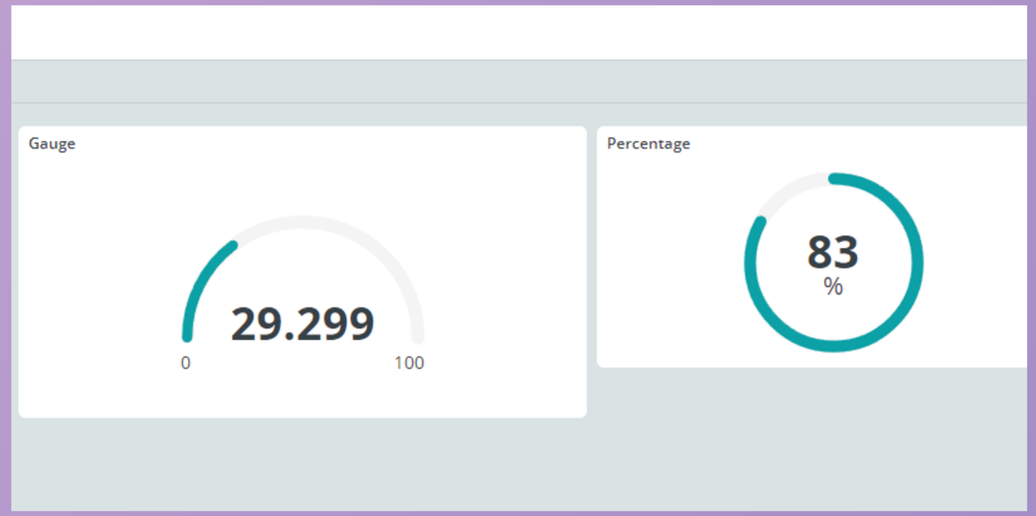
```
#include <DHT.h>
#define DHTPIN D5
#define DHTTYPE    DHT11

DHT dht(DHTPIN, DHTTYPE);
#include "thingProperties.h"
int ledPin = D4;

void setup() {
  // Initialize serial and wait
  dht.begin();
  Serial.begin(9600);
  // This delay gives the chance
  pinMode(ledPin, OUTPUT);
  delay(1500);
  digitalWrite(ledPin, LOW);
```

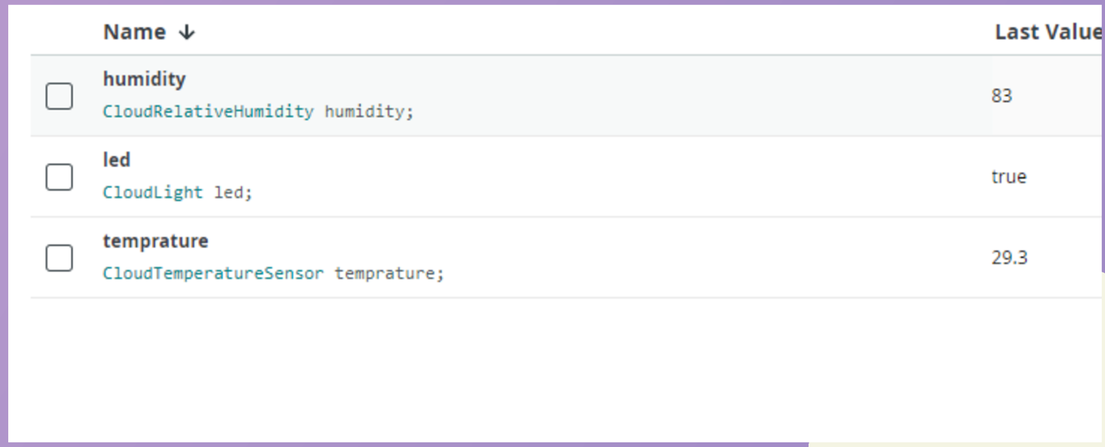
SETTING UP DASHBOARD

Set up two variables in the Arduino IoT Cloud for temperature and humidity.

Then, create a gauge for temperature and another one for humidity inside the dashboard.

After that We will start working on the code to get our value





OVER VIEW AND EXPLANATION OF THE CODE

HOME_AUTOMATION_aug15 | ReadMe.adoc | thingProperties.h | Secrets Tab

```
#include <DHT.h>
#define DHTPIN D5
#define DHTTYPE    DHT11

DHT dht(DHTPIN, DHTTYPE);
#include "thingProperties.h"
int ledPin = D4;

void setup() {
  // Initialize serial and wait for port to open:
  dht.begin();
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  pinMode(ledPin, OUTPUT);
  delay(1500);
  digitalWrite(ledPin, LOW);



  // 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
  temprature = dht.readTemperature();
  humidity = dht.readHumidity();


}


/*
  Since Led is READ_WRITE variable, onLedChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLedChange()  {
  // Add your code here to act upon Led change
  if(led == 1){
```

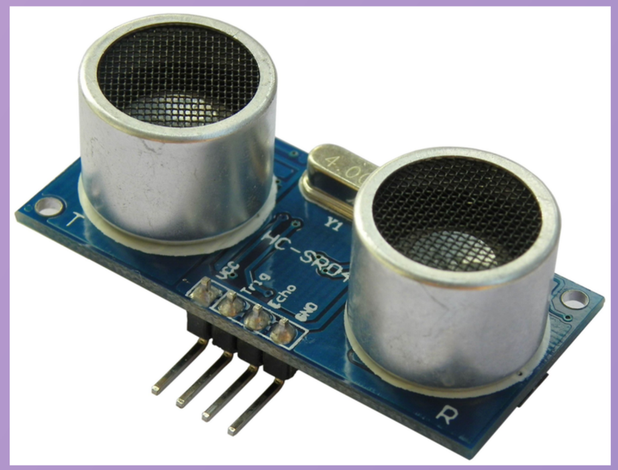
YOU JUST COMPLETED THE FIRST PROJECT

THE HOME SECURITY SYSTEM

BE READY TO IMPORVE THE DOOR SECURITY SYSTEM YOU ALREADY MADE

THE APPROACH

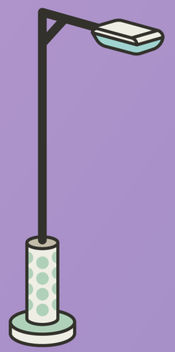
Ultrasonic Sonar is utilized for measuring the distance between two points.



Rather than the IR sensor you utilized, let's switch to a Sonar to add a touch of excitement.

Let us look a bit to the Sonar Sensor

HOW DOES A SONAR WORK



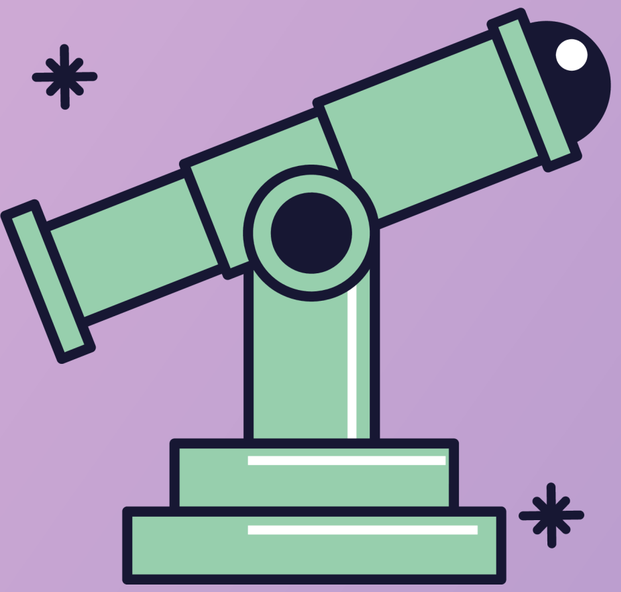
SONAR CONSIST OF A TRANSMITER AND A RECIEVER

DURATION = DISTANCE * 0.034/2

The transistor emits an ultrasonic sound, while the receiver picks up the sound waves that reflect back after colliding with any material.

THE DISTANCE IS THEN CALCULATED BY USING THE EQUATIONS SHOWN

LET'S NOW PROCEED WITH SETTING UP A MONITORING SYSTEM USING OUR SONAR.



FIRST THINGS FIRST PROCEED WITH CREATING A THING AND SETTING UP THE VARIABLES (A DISTANCE VARIABLE)

Remember to save the security files as they are crucial.

Now that the environment is set up, let's start working on our sketch.

READING DATA OUT OF A SONAR

THE ABOCE PIECE OF CODE IS USED TO READ DATA FROM A SONAR SENSOR. FROM THIS POINT TRY ON YOU OWN TO SETUP A DASH BOARD AND SET DATA TO YOUR VARIABLE

MAKE THE LOGIC LIKE IF THE DISTANCE COMES UNDER 10 THE LED TRUNS ON OR THE BUZZER IS TRIGGERED

```
int trigPin = D6;
int echoPin = D7;
int ledPin = D8;
bool ledState = false;
void setup() {
  // Initialize serial and wait for port to op
  Serial.begin(9600);
  // This delay gives the chance to wait for a
  delay(1500);
  pinMode(trigPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnec
  digitalWrite(trigPin, LOW);
  /*
```

```
int distMeasure(){
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);


  long duration = pulseIn(echoPin, HIGH);

  float distanceMeters = (duration * 0.0343) / 2.0;

  distance = distanceMeters;
  Serial.println(distanceMeters);
  delay(1000);
  return distanceMeters;
}
void loop() {
  ArduinoCloud.update();
  int dist = distMeasure();
```

YOU GUYS JUST UPDATED A SYSTEM YOU ALREADY MADE

CONGRATULATIONS

THANKYOU

MADE BY HARIKESH OP
S3 ECA GCEK