Things used in this project

Hardware components

National Control Devices ESP-32× 1

National Control Devices IoT Long Range Wireless Temperature And Humidity Sensor:× 1

Software apps and online services

Arduino IDE

Amazon Web Services AWS IoT

Story:

Hardware

* **ESP-32:**The ESP32 makes it easy to use the Arduino IDE and the Arduino Wire Language for IoT applications. This ESP32 IoT module combines Wi-Fi, Bluetooth, and Bluetooth BLE for a variety of diverse applications. This module comes fully-equipped with 2 CPU cores that can be controlled and powered individually, and with an adjustable clock frequency of 80 MHz to 240 MHz. This ESP32 IoT WiFi BLE Module with Integrated USB is designed to fit in all ncd.io IoT products. Monitor sensors and control relays, FETs, PWM controllers, solenoids, valves, motors and much more from anywhere in the world using a web page or a dedicated server. We manufactured our own version of the ESP32 to fit into NCD IoT devices, offering more expansion options than any other device in the world! An integrated USB port allows easy programming of the ESP32. The ESP32 IoT WiFi BLE Module is an incredible platform for IoT application development. This ESP32 IoT WiFi BLE Module can be programmed using the Arduino IDE.
* [**IoT Long Range Wireless Temperature and Humidity Sensor**:](https://store.ncd.io/product/industrial-long-range-wireless-temperature-humidity-sensor/) Industrial Long Range Wireless Temperature Humidity Sensor. Grade with a Sensor Resolution of ±1.7%RH ±0.5° C.Up to 500, 000 Transmissions from 2 AA Batteries.Measures -40°C to 125°C with Batteries that Survive these Ratings.Superior 2-Mile LOS Range & 28 miles with High-Gain Antennas.Interface to Raspberry Pi, Microsoft Azure, Arduino and More
* [**Long-Range Wireless Mesh Modem with USB Interface**](https://store.ncd.io/product/900hp-s3b-long-range-wireless-mesh-modem-with-usb-interface/)

### Software Used

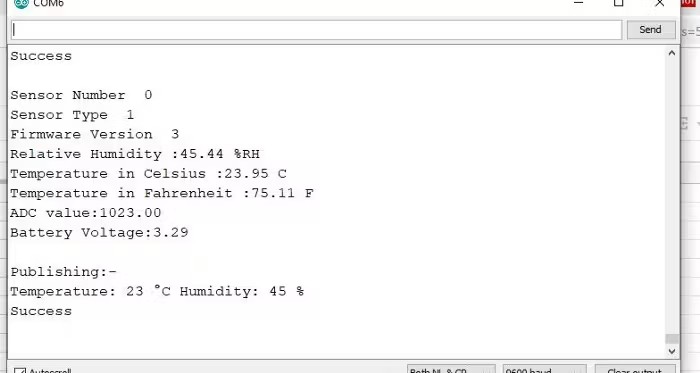
* Arduino IDE
* AWS

### Library Used

* PubSubClient Library
* Wire.h
* AWS\_IOT.h

### Uploading the Code to ESP32 Using Arduino IDE

* Download and include the PubSubClient Library and Wire.h Library.
* Download the Zip file of AWS\_IoT, from the given [link](https://github.com/ExploreEmbedded/Hornbill-Examples)and after extracting, paste the library in your Arduino library folder.
* You must assign your unique AWS MQTT\_TOPIC, AWS\_HOST, SSID (WiFi Name) and Password of the available network.
* MQTT topic and AWS HOST can get inside Things-Interact at AWS-IoT console.
* Compile and upload the [ESP32\_AWS.ino](https://github.com/mjScientech/Monitoring-Temp-and-Humidity-using-AWS-ESP32/blob/master/ESP32_AWS.ino)code.
* To verify the connectivity of the device and the data sent, open the serial monitor. If no response is seen, try unplugging your ESP32 and then plugging it again. Make sure the baud rate of the Serial monitor is set to the same one specified in your code 115200.

Serial Monitor Output

Making the AWS Work

CREATE THING AND CERTIFICATE

THING: It is a virtual representation of your device.

CERTIFICATE: Authenticates the identity of a THING.

Open AWS-IoT

Click on manage -THING -Register THING.