```
/****************************************
 * Include Libraries
 ****************************************/
#include <WiFi.h>
#include <PubSubClient.h>

#define WIFISSID "" // Put your WifiSSID here
#define PASSWORD "" // Put your wifi password here
#define TOKEN "xxxxx" // Put your Ubidots' TOKEN
#define MQTT_CLIENT_NAME "xxxx" // MQTT client Name, please enter your own 8-12
alphanumeric character ASCII string;
                                            //it should be a random and unique ascii
string and different from all other devices

/****************************************
 * Define Constants
 ****************************************/
#define VARIABLE_LABEL "Demo" // Assing the variable label
#define DEVICE_LABEL "Demo" // Assig the device label

#define SENSOR_PIN 36

char mqttBroker[]  = "industrial.api.ubidots.com";
char payload[100];
char topic[150];
// Space to store values to send
char str_sensor[10];

/****************************************
 * Auxiliar Functions
 ****************************************/
WiFiClient ubidots;
PubSubClient client(ubidots);

void callback(char* topic, byte* payload, unsigned int length) {
  char p[length + 1];
  memcpy(p, payload, length);
  p[length] = NULL;
  String message(p);
  Serial.write(payload, length);
  Serial.println(topic);
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.println("Attempting MQTT connection...");

    // Attemp to connect
    if (client.connect(MQTT_CLIENT_NAME, TOKEN, "")) {
      Serial.println("Connected");
    } else {
      Serial.print("Failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 2 seconds");
      // Wait 2 seconds before retrying
      delay(2000);
    }
  }
```

```
}

/**************************************
 * Main Functions
 **************************************/
void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFISSID, PASSWORD);
  // Assign the pin as INPUT
  pinMode(SENSOR_PIN, INPUT);

  Serial.println();
  Serial.print("Wait for WiFi...");

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  client.setServer(mqttBroker, 1883);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }

  sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
  sprintf(payload, "%s", ""); // Cleans the payload
  sprintf(payload, "{\"%s\":", VARIABLE_LABEL); // Adds the variable label

  float sensor_data = analogRead(SENSOR_PIN);
  Serial.println(sensor_data);
  /* 4 is mininum width, 2 is precision; float value is copied onto str_sensor*/
  dtostrf(sensor_data, 4, 2, str_sensor);

  sprintf(payload, "%s {\"value\": %s}}", payload, str_sensor); // Adds the value
  Serial.println("Publishing data to Ubidots Cloud");
  client.publish(topic, payload);
  client.loop();
  delay(1500);
}
```