



Amirkabir University of Technology  
(Tehran Polytechnic)



# مزرعه هوشمند

علی انتظاری، سینا ربیعی، محمد مسیح شالچیان ناظر، سید رضا هندی

دکتر افشار

زمستان ۱۴۰۰

## فهرست مطالب

۲

## ۱. المان‌ها و اجزای سخت‌افزاری مدار

۱,۱. نام، شرکت سازنده، قیمت و مشخصات اصلی..... ۲

۱,۲. برشور و کاتالوگ..... ۷

۱,۳. گزارش ساخت..... ۷

۱,۴. گزارش تست..... ۷

۷

## ۲. کد سیستم و نرم‌افزارهای استفاده شده

۲,۱. اصل کد و توضیحات مربوطه..... ۷

۲,۲. پلتفرم ThingsBoard..... ۱۲

a) معرفی کلی پلتفرم ThingsBoard:..... ۱۲

b) شروع به کار با ThingsBoard:..... ۱۳

۲,۳. Arduino IDE..... ۱۵

۲,۴. محل کاربری سیستم..... ۱۶

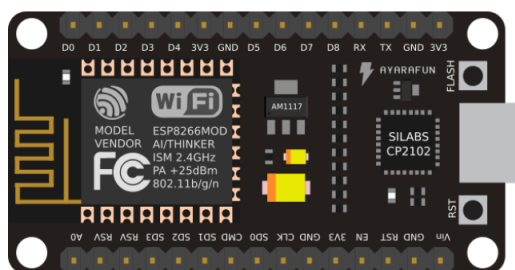
۲,۵. صفحات نشان‌داده شده به کاربر..... ۱۶

## ۱. المان‌ها و اجزای سخت‌افزاری مدار

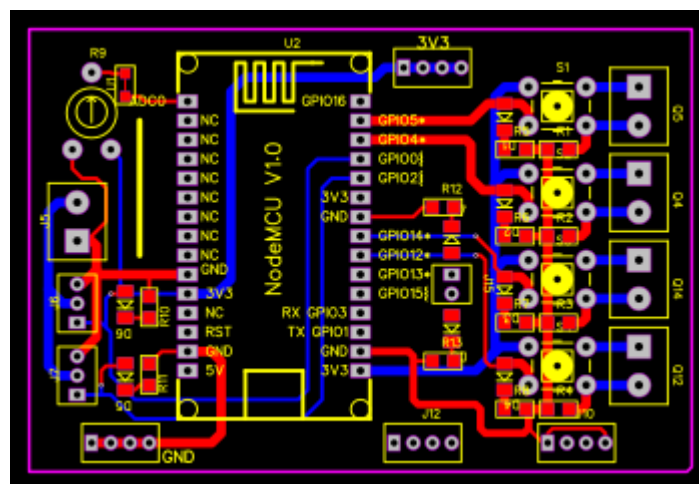
۱.۱. نام، شرکت سازنده، قیمت و مشخصات اصلی

(a) برد NodeMCU:

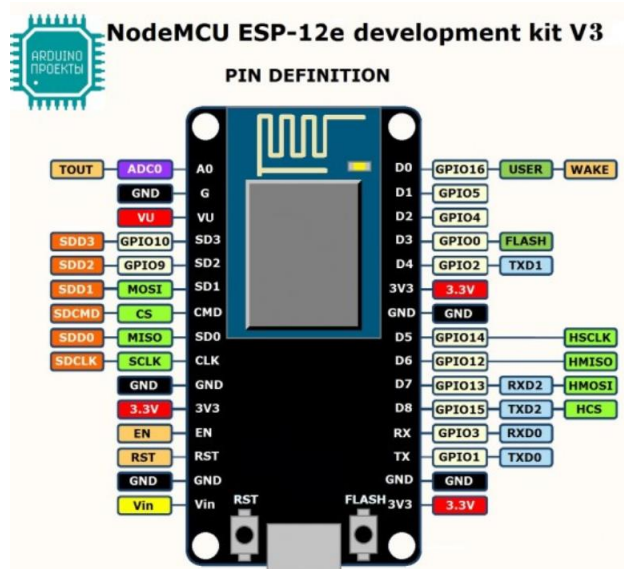
این برد بر پایه ماژول ESP8266 بنا شده و حاوی این ماژول به همراه تعدادی واحد بیشتر ADC، GPIO و ... می باشد. NodeMCU بر اساس زبان اسکریپتی Lua ساخته شده؛ اما می توان کد های آردوینو (زبان C) را نیز توسط محیط آردوینو بر آن برنامه ریزی کرد. در واقع استفاده از این برد کار با ماژول ESP8266 را ساده تر می کند و امکانات بیشتری در اختیار می گذارد. در ادامه شکاتیک کلی برد و مشخصات پین‌های آن را می‌بینیم.



شکل ۱: نمای کلی برد NodeMCU



شکل ۲: نمای PCB برد NodeMCU



شکل ۳: پین‌های برد NodeMCU

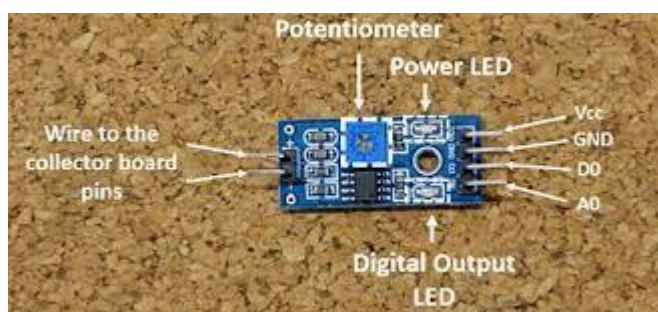
(b) سنسور رطوبت خاک (YL69) :

همانطور که از نام این سنسور مشخص است، برای اندازه‌گیری میزان رطوبت و یا آب موجود در خاک و زمین استفاده می‌گردد با استفاده از این سنسور، برد آردوینو، یک رله و یک پمپ آب ساده همه چیز برای ساخت یک سیستم آبیاری هوشمند برای گلدان منزلتان فراهم است، خروجی آنالوگ این سنسور در زمانی که میزان رطوبت خاک کم باشد، مقدار خروجی بالا و زمانی که رطوبت بالا باشد مقدار پایین‌تری را به پین آنالوگ آردوینو ارسال می‌نماید. از این سیستم می‌توان برای ساخت گلخانه‌های هوشمند، سیستم‌های آبیاری خودکار زمین‌های کشاورزی و باغ‌ها، آبیاری هوشمند گلدان‌های آپارتمانی و ... استفاده کرد. در این پروژه از سنسور ساخته شده توسط شرکت EVANA استفاده شده. لازم به توجه است که این سنسور توسط شرکت‌های فراوانی ساخته شده که نحوه کار و مشخصات آن‌ها یکسان می‌باشد. قیمت این سنسور ۱۵ هزار تومان می‌باشد. از مشخصات این سنسور می‌توان به موارد زیر اشاره کرد:

- حساسیت قابل تنظیم از طریق پتانسیومتر آبی رنگ موجود بر روی برد (برای مشخص کردن مقدار آستانه دیجیتال)
- ولتاژ عملیاتی بین ۳٫۳ تا ۵ ولت
- دارای خروجی دیجیتال و آنالوگ
- LED نماینگر پاور (سبز)
- چیپ مقایسه گر LM393 با پایداری بالا
- AO خروجی آنالوگ (۰ تا ۴٫۲ ولت)
- DO خروجی دیجیتال (۰ یا یک)



شکل ۴: نمای کلی برد سنسور رطوبت خاک YL69



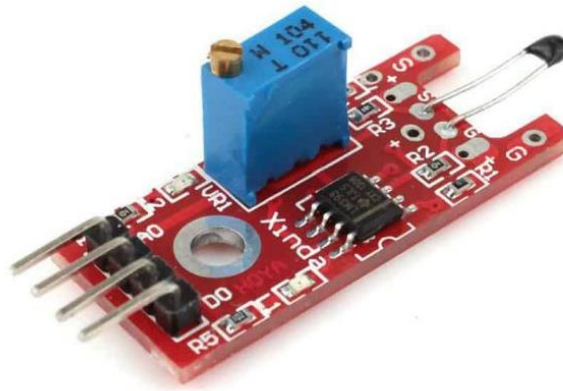
شکل ۵: نمای کلی برد سنسور رطوبت خاک YL69

### (C) سنسور دما (KY-028) :

سنسور دما دیجیتال KY-028 برای آردوینو، تغییرات دما را بر اساس تغییرات مقاومت ترمیستور اندازه‌گیری می‌کند. این ماژول دارای دو خروجی دیجیتال و آنالوگ است و یک پتانسیومتر برای تنظیم آستانه تشخیص در رابط دیجیتال وجود دارد. این سنسور در صفحه مدار خود دارای ۳ مؤلفه اصلی است. ابتدا واحد سنسور در قسمت جلویی یک سیگنال آنالوگ را به واحد دوم، یعنی تقویت‌کننده می‌فرستد. آمپلی‌فایر با توجه به مقدار مقاومت پتانسیومتر، سیگنال را تقویت می‌کند و سیگنال را به خروجی آنالوگ ماژول می‌فرستد. مؤلفه سوم مقایسه‌گری است که اگر سیگنال تحت یک مقدار خاص قرار گیرد، خروجی دیجیتال و LED را خاموش می‌کند (با تنظیم پتانسیومتر می‌توانید حساسیت را کنترل کنید). قیمت این ماژول ۲۵ هزار تومان می‌باشد. از مشخصات این سنسور می‌توان به موارد زیر اشاره کرد:

- دارای ترمیستور NTC و بازه اندازه‌گیری ۵۵- تا ۱۲۵ درجه سانتی‌گراد
- یک مقایسه‌گر دیفرانسیل LM393 دوتایی

- پتانسیومتر 3296W
- شش مقاومت
- دارای خروجی آنالوگ و دیجیتال



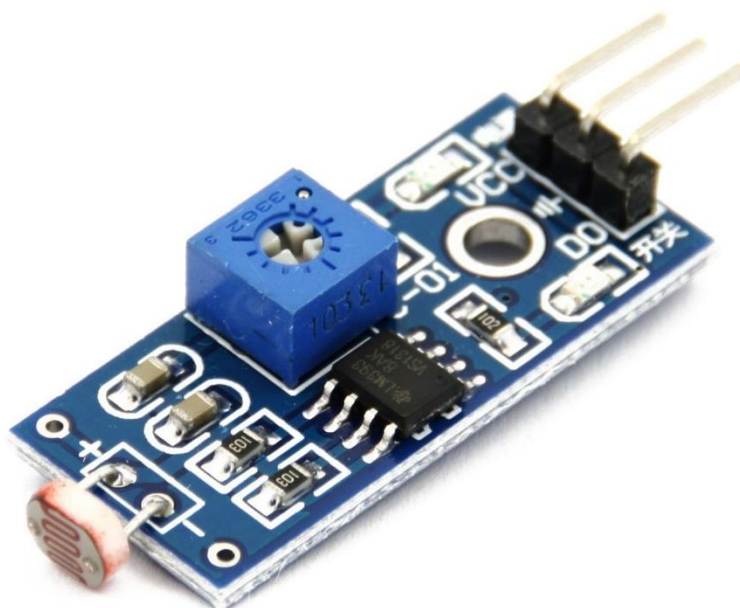
شکل ۶: نمای کلی برد سنسور دما KY-028

(d) سنسور نور (SYN-610300) :

سنسور نور با خروجی دیجیتال ساخت شرکت SunRobotics، بر اساس تغییرات مقاومت (Photocell) اندازه‌گیری می‌کند. بر روی این ماژول یک عدد فتوسل به همراه مدار مقایسه کننده و یک پتانسیومتر وجود دارد. مقاومت سنسور فتوسل متناسب با نور دریافت شده توسط آن تغییر می‌کند و در نتیجه تغییر مقاومت، خروجی مقایسه کننده، با مقایسه این مقدار با مقدار تعیین شده توسط پتانسیومتر تغییر وضعیت خواهد داد. ماژول دارای خروجی دیجیتال می‌باشد که قابلیت اتصال آن به میکروکنترلر و آردوینو و ماژول رله را فراهم می‌کند. از این ماژول در کنترل لامپ‌های روشنایی معابر و ساختمان‌ها (جهت کاهش مصرف انرژی استفاده می‌شود)، در خطوط تولید (سنسور در مقابل یک لامپ قرار می‌گیرد و عبور هر کالا را جهت شمارش تولید یا بسته‌بندی در یک تعداد مشخص و یا حتی جهت کنترل کیفیت محصول تولیدی)، در رباتیک (تشخیص موانع و یا تعقیب خطوط) استفاده می‌شود. معمولترین ماده‌ای که در این سنسور نوری استفاده می‌شود Cadmium sulphide است. این ماده در شکل زیر نشان داده شده است. سعی کنید هر وقت خواستید از فتوسل استفاده کنید سطح بیرونی آن رو تمیز کنید تا بطور کامل در معرض نور قرار بگیرد وگرنه اگر سطح خارجی با گردوخاک پوشیده شده باشد نور رسیده به آن کم می‌شود و مقاومت آن زیاد می‌شود. قیمت این ماژول ۱۶ هزار تومان می‌باشد. از مشخصات این سنسور می‌توان به موارد زیر اشاره کرد:

- اندازه کوچک (۳ در ۱,۶ سانتیمتر)
- چیپ مقایسه گر LM393
- شش مقاومت

- LED نماینگر پاور و خروجی دیجیتال (هر دو سبز)
- دیجیتال (تنظیم آستانه با پتانسیومتر)



شکل ۷: نمای کلی برد سنسور نور (LDR) SYN-610300

## ۱.۲. برشور و کاتالوگ

کاتالوگ‌های مربوط به این چهار قطعه معرفی شده در بالا، در فایل تحویل داده شده برای پروژه موجود می‌باشد.

## ۱.۳. گزارش ساخت

همانطور که از تصاویر مشخص است، برد و تمامی سنسورها روی یک Breadboard تعبیه شدند و ارتباط آن‌ها با کامپیوتر از طریق کابل Micro USB برقرار می‌باشد. همه‌ی سنسورها از منبع تغذیه ۳،۳ ولت برد NodeMCU استفاده می‌کنند و زمین آن‌ها نیز همان زمین ارائه شده توسط برد می‌باشد. توجه داریم که چون جریان‌های تولید شده در این مدار در حد ۲ میلی‌آمپر و پایین‌تر می‌باشد، نیازی به جدا کردن زمین‌ها و استفاده از تکنیک‌های shielding و grounding نیست.

## ۱.۴. گزارش تست

با توجه به ویدیوهای موجود در فایل‌های ارسال شده، مشاهده می‌شود که اجزای مدار در حالت مطلوب عمل می‌کنند. توجه داریم که از خروجی دیجیتال سنسور نور استفاده شده و آستانه آن را می‌توان با پتانسیومتر موجود روی برد تغییر داد.

## ۲. کد سیستم و نرم افزارهای استفاده شده

### ۲.۱. اصل کد و توضیحات مربوطه

```

1.const unsigned int Light_PIN = D1;
2.const unsigned int Moisture_PIN = D2;
3.const unsigned int Temp_Analog = A0;
4.
5.void setup(){
6.  Serial.begin(9600);
7.  pinMode(Light_PIN, INPUT);
8.  pinMode(Moisture_PIN, INPUT);
9.  pinMode(Temp_Analog, INPUT);
10. }
11.
12. void loop(){
13.  //Temperature Sensor
14.  int temprature = analogRead (Temp_Analog);
15.  //Moisture Sensor
16.  int Moisture = digitalRead(Moisture_PIN);
17.  //Light Blocking
18.  int LDR = digitalRead(Light_PIN);
19.  //Serial.println("LDR:");
20.  Serial.println(LDR);
21.  //Serial.println("Moisture:");
22.  Serial.println(Moisture);
23.  //Serial.println("Temp:");
24.  Serial.println(temprature);
25.  delay(500);
26. }

```

کد شماره‌ی ۱: دریافت اطلاعات در Arduino IDE

همانطور که در کد بالا دیده می‌شود، ابتدا پین‌های بصورت متغیرهای ثابت تعریف شده‌اند. سپس در قسمت **setup** ریت کاری برد را روی ۹۶۰۰ تنظیم کرده (دقت داریم که این ریت باید با ریت **Serial Monitor** برابر باشد تا داده‌ها به درستی نمایش داده شود) و نوع پین‌های تعریف شده را مشخص می‌کنیم که در کاربرد ما همگی ورودی می‌باشند. در تابع **loop** موجود در کد شماره‌ی ۱ داده‌ها هر ۵۰۰ میلی ثانیه دریافت می‌شوند و در **Serial Monitor** نمایش داده می‌شود.

```
1.from runpy import _TempModule
```



```
2.import paho.mqtt.client as paho
3.import os
4.import json
5.import time
6.import serial
7.import schedule
8.import random
9.from datetime import datetime
10.
11. ACCESS_TOKEN = "ESP8266"
12. broker = "demo.thingsboard.io"
13. port = 1883
14.
15. client1 = paho.Client("controll")
16. client1.username_pw_set(ACCESS_TOKEN)
17. client1.connect(broker, port, keepalive=60)
18.
19.
20. def main_func():
21.     arduino = serial.Serial("COM5", 9600)
22.     arduino_data_LDR = arduino.readline()
23.     light = str(arduino_data_LDR[0 :
        len(arduino_data_LDR)].decode("utf-8"))[:-1]
24.     arduino_data_Mositure = arduino.readline()
25.     Moisture = str(arduino_data_Mositure[0:
        len(arduino_data_Mositure)].decode("utf-8")
26.     )[:-1]
27.     arduino_data_Temp = arduino.readline()
28.     Temp = str(arduino_data_Temp[0 :
        len(arduino_data_Temp)].decode("utf-8"))[:-1]
29.
30.     #####
31.     payload = "{"
32.     payload += '"temperature":' + Temp + ","
33.     payload += '"moisture":' + Moisture + ","
34.     payload += '"light":' + light
35.     payload += "}"
36.     ret =
        client1.publish("v1/devices/me/telemetry", payload)
```

```

37.     print(payload)
38.     #####
39.
40. list_values = []
41. list_in_floats = []
42.
43. print("Program started")
44. # Setting up the Arduino
45. schedule.every(1).seconds.do(main_func)
46.
47. while True:
48.     schedule.run_pending()
49.     time.sleep(0.5)

```

کد شماره‌ی ۲: ارسال اطلاعات به آدرس مورد نظر توسط Python

در کد شماره‌ی ۲ داده‌های دریافت شده توسط آردوینو روی پورت مورد نظر توسط کتابخانه **serial** پایتون با ریت مشخص شده دریافت می‌شود و در یک فرمت تعیین شده به سرور فرستاده می‌شود. این کد از پروتکل **Mqtt** بر پورت ۱۸۸۳ استفاده می‌کند. پارامترهایی که برای کاربران مختلف ممکن است نیاز به تغییر داشته باشند:

broker	آدرس سایتی که داده‌ها به آن ارسال می‌شود
ACCESS_TOKEN	توکن تعریف شده در پلتفرم برای دسترسی به داشبورد

```

1. #include <PubSubClient.h>
2. #include "ThingsBoard.h"
3. #include <ESP8266WiFi.h>
4.
5. #define WIFI_AP "SinaR"
6. #define WIFI_PASSWORD "0420sina"
7. #define TOKEN "ESP8266"
8. #define THINGSBOARD_SERVER "192.168.43.199"
9.
10. WiFiClient espClient;
11. ThingsBoard tb(espClient);
12. int status = WL_IDLE_STATUS;
13.
14. void setup()
15. {

```

```
16.   Serial.begin(9600);
17.   WiFi.begin(WIFI_AP, WIFI_PASSWORD);
18.   InitWiFi();
19. }
20.
21. void loop()
22. {
23.   delay(500);
24.
25.   if (WiFi.status() != WL_CONNECTED) {
26.     reconnect();
27.   }
28.   if (!tb.connected()) {
29.     Serial.print("Connecting to: ");
30.     Serial.print(THINGSBOARD_SERVER);
31.     Serial.print(" with token ");
32.     Serial.println(TOKEN);
33.     if (!tb.connect(THINGSBOARD_SERVER,TOKEN)) {
34.       Serial.println("Failed to connect");
35.       return;
36.     }
37.   }
38.
39.   Serial.println("Sending data...");
40.   int temprature = analogRead (Temp_Analog);
41.   int Moisture = digitalRead(Moisture_PIN);
42.   int LDR = digitalRead(Light_PIN);
43.   tb.sendTelemetryInt("temperature", temprature);
44.   tb.sendTelemetryInt("moisture", Moisture);
45.   tb.sendTelemetryInt("light", LDR);
46.   tb.loop();
47. }
48.
49. void InitWiFi()
50. {
51.   Serial.println("Connecting to AP ...");
52.   WiFi.begin(WIFI_AP, WIFI_PASSWORD);
53.   while (WiFi.status() != WL_CONNECTED) {
54.     delay(500);
```

```

55.     Serial.print(".");
56. }
57.     Serial.println("Connected to AP");
58. }
59.
60. void reconnect() {
61.     status = WiFi.status();
62.     if ( status != WL_CONNECTED) {
63.         WiFi.begin(WIFI_AP, WIFI_PASSWORD);
64.         while (WiFi.status() != WL_CONNECTED) {
65.             delay(500);
66.             Serial.print(".");
67.         }
68.         Serial.println("Connected to AP");
69.     }
70. }

```

کد شماره‌ی ۳: دریافت اطلاعات در Arduino IDE و ارسال به Localhost

در این کد داده‌های دریافت شده توسط آردوینو به طور مستقیم به localhost ارسال می‌شوند. در این بخش لازم است که از اتصال برد به شبکه داخلی مطمئن شویم که این موضوع توسط دو تابع InitWiFi و reconnect انجام می‌شود. پس از برقراری ارتباط، داده توسط کتابخانه ThingsBoard.h به شبکه داخلی فرستاده می‌شود.

broker	Localhost IP مورد نظر
ACCESS_TOKEN	توکن تعریف شده در پلتفرم برای دسترسی به داشبورد
WIFI_AP	نام شبکه داخلی
WIFI_PASSWORD	پسورد شبکه داخلی

لازم به ذکر است که کدهای نوشته شده در آردوینو، از زبان C می‌باشد که برای راحتی به تعدادی کتابخانه کاربردی مجهز شده.

## ۲.۲. پلتفرم ThingsBoard

(a) معرفی کلی پلتفرم ThingsBoard:

بخش‌های اصلی اکثر کاربردهای اینترنت اشیاء شامل جمع‌آوری داده‌ها، ارسال داده‌ها، نمایش داده‌ها و عملیات متناسب می‌باشد. یکی از بخش‌هایی که در همه این کاربردها یکسان است در واقع دریافت داده‌های ارسال شده توسط یک وب سرور و نمایش آن به کاربر می‌باشد. در این مرحله باید اطلاعات اولاً به درستی دریافت شوند و

ثانیا به زیبایی نمایش داده شوند. همچنین چالش دیگر این قسمت نمایش زنده داده‌ها است؛ بدین معنی که به محض تغییر در کمیت‌های اندازه‌گیری شده باید این تغییرات به کاربر نمایش داده شوند. پلتفرم‌های مختلفی توسط شرکت‌های معتبر جهت برطرف کردن نیازهای این بخش توسعه یافته اند که از جمله آن‌ها می‌توان به پلتفرم‌های معروف ThingsBoard و ThingsSpeak اشاره کرد.

به کمک ThingsBoard می‌توان به راحتی داده‌های سنسورها و دستگاه‌های مختلف را دریافت کرد و با نمودارها و قالب‌های فراوان آماده این پلتفرم به زیبایی هر چه تمام‌تر آن‌ها را نمایش داد. همچنین امکان معرفی کاربرهای مختلف، تعیین رمز برای کاربرها و سطح دسترسی هر کاربر به داده‌ها به سادگی توسط رابط گرافیکی<sup>۱</sup> این پلتفرم وجود دارد. بدین معنی که هر کاربر تنها داده‌هایی را مشاهده خواهد کرد که توسط مدیر پلتفرم تعیین شده است. همچنین این پلتفرم قابلیت تعریف روابط علل و معلولی را دارد. در نتیجه با توجه به داده‌های دریافت شده می‌توان تصمیم لازم را گرفت و عملگرهای لازم را به کار برد. در شکل ۸ کاربرد ThingsBoard در یک مزرعه هوشمند را می‌بینیم.



شکل ۸: مزرعه هوشمند در Things board

## (b) شروع به کار با ThingsBoard:

در این قسمت مراحل لازم برای شروع به کار با ThingsBoard را بررسی می‌کنیم. این مراحل در تمامی پروژه‌های نوشته شده مشترک هستند.

### ۱. آماده‌سازی دستگاه<sup>۲</sup>:

در این مرحله دستگاه مورد استفاده خود (آردوینو یا ESP8266 یا هر دستگاه دیگر مورد استفاده) را به ThingsBoard معرفی و متصل می‌کنیم. برای این کار در رابط گرافیکی<sup>۳</sup> ThingsBoard به قسمت Devices رفته و بر روی دکمه "+" کلیک می‌کنیم. در پنجره باز شده مطابق شکل ۹ نام دلخواه خود را به

<sup>1</sup> GUI

<sup>2</sup> Provision your device

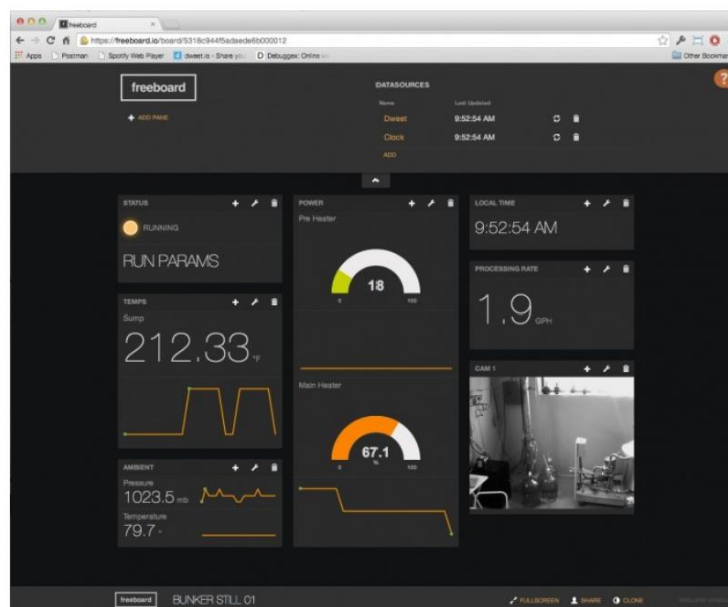
<sup>3</sup> این قسمت در شبکه‌های محلی از آدرس <http://localhost:8080> قابل دسترسی است.

دستگاه می‌دهیم. پس از معرفی دستگاه، وارد قسمت **Manage credentials** مربوط به آن دستگاه می‌شویم. جزییات دستگاه در این قسمت موجود است. در این قسمت **Access Token** مربوط به دستگاه را یادداشت می‌کنیم. در کدهای مربوط به میکروهای مختلف برای برقراری ارتباط بین دستگاه و **ThingsBoard** به این **Access Token** نیاز داریم.

شکل ۹: آماده‌سازی دستگاه

## ۲. ساخت داشبورد:

داشبورد در واقع قسمت قابل مشاهده و گرافیکی از داده‌های جمع آوری شده‌است. همچنین **ThingsBoard** این قابلیت را دارد که به هر کاربر داشبورد مخصوص خود را نمایش دهد؛ به این معنی که نوع داده‌های نمایش داده شده و نحوه نمایش داده‌ها (نوع نمودار و ...) می‌تواند برای هر کاربر با کاربر دیگر متفاوت باشد. جهت ساخت داشبورد در قسمت **Dashboard** بر روی دکمه "+" کلیک کنید و سپس وارد قسمت **Add new dashboard** شده و مشخصات داشبورد مورد نظر خود را وارد کنید. در تصاویر زیر نمونه‌ای از داشبوردهای ساخته شده با این پلتفرم را می‌بینیم.



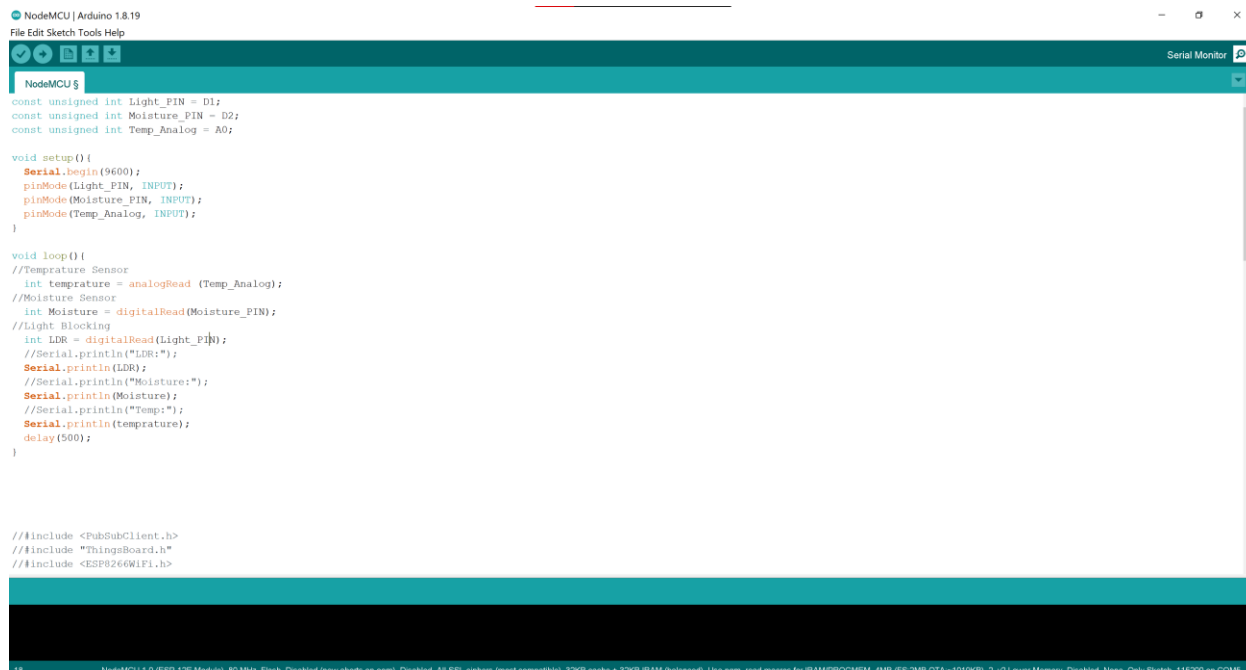
شکل ۱۰: نمونه‌ای از داشبورد ساخته شده در Things board

در این مرحله دستگاه مورد استفاده خود (آردوینو یا ESP8266 یا هر دستگاه دیگر مورد استفاده) را به ThingsBoard معرفی و متصل می‌کنیم. برای این کار در رابط گرافیکی<sup>۴</sup> ThingsBoard به قسمت Devices رفته و بر روی دکمه "+" کلیک می‌کنیم. در پنجره باز شده مطابق شکل ۹ نام دلخواه خود را به دستگاه می‌دهیم. پس از معرفی دستگاه، وارد قسمت Manage credentials مربوط به آن دستگاه می‌شویم. جزییات دستگاه در این قسمت موجود است. در این قسمت Access Token مربوط به دستگاه را یادداشت می‌کنیم. در کدهای مربوط به میکروهای مختلف برای برقراری ارتباط بین دستگاه و ThingsBoard به این Access Token نیاز داریم.

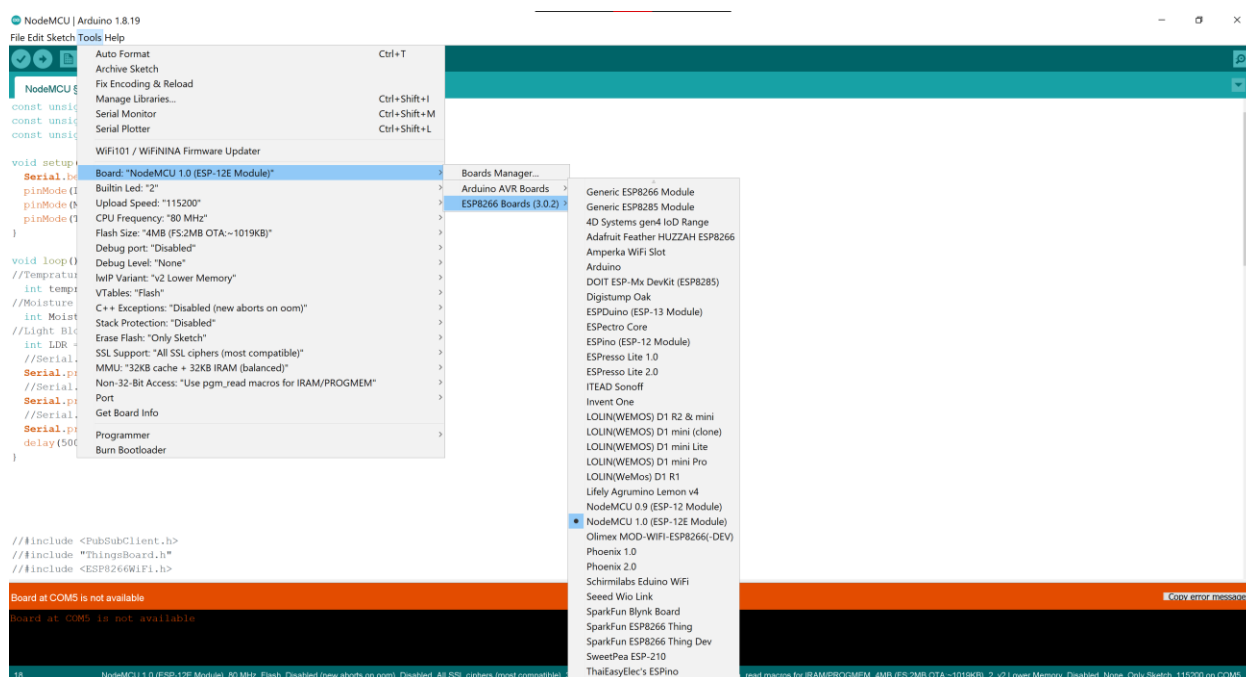
### ۲.۳. Arduino IDE

این IDE همانند IDE های دیگر، یک محیط طراحی شده برای کد زدن با راحتی بیشتر می‌باشد. از امکانات این IDE می‌توان به مثال‌های فراوان که با اضافه کردن هر کتابخانه به منوهای اضافه می‌شود اشاره نمود. همچنین این IDE دارای یک بخش اختصاصی به serial monitor می‌باشد که امکان بررسی داده‌های دریافت و ارسال شده توسط برد را فراهم می‌کند. در منوهای مختلف این نرم‌افزار امکان انتخاب برد مورد نظر از رنج وسیعی از بردهای Arduino و... موجود می‌باشد. در ادامه تصاویری از محیط این IDE را می‌بینیم:

<sup>۴</sup> این قسمت در شبکه های محلی از آدرس <http://localhost:8080> قابل دسترس است.



شکل ۱۱: نمای Arduino IDE



شکل ۱۲: بردهای موجود در Arduino IDE

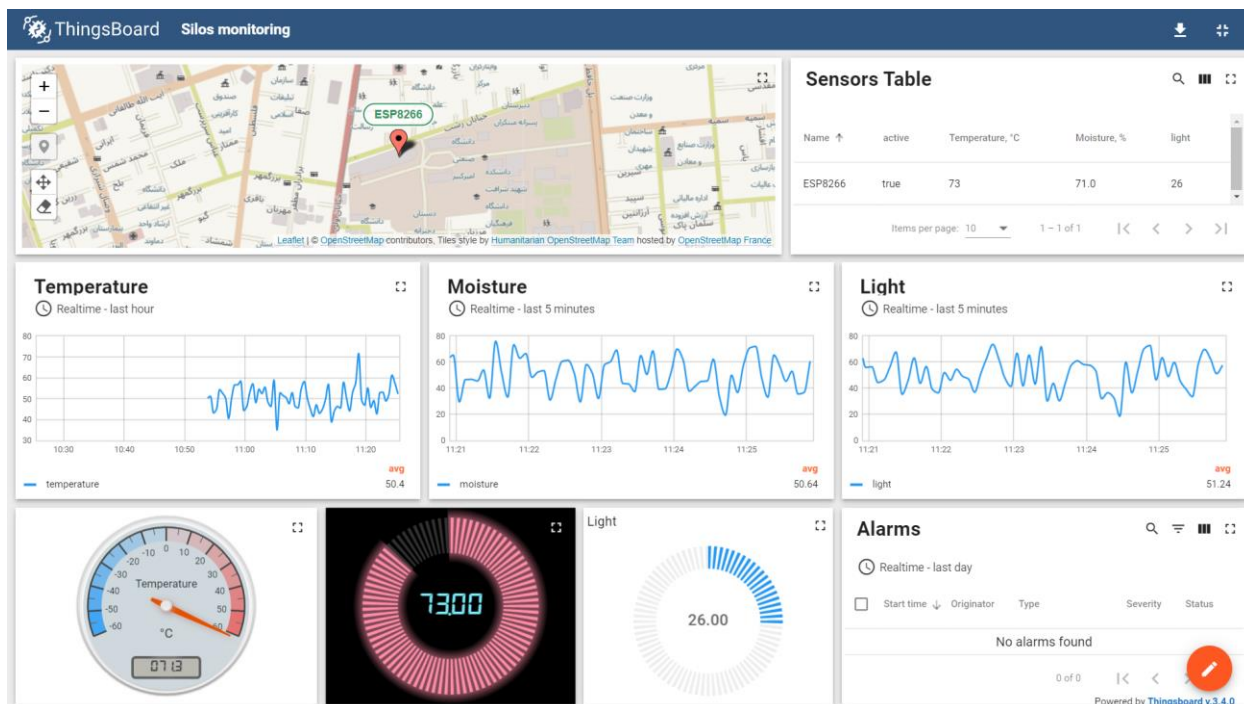
## ۲.۴. محل کاربری سیستم

سیستم طراحی شده برای کنترل و بررسی وضعیت پارامترهای حیاتی گیاه استفاده می‌شود. با توجه به این موضوع، نیازی به کاربر برای کارکرد آن نیست. برد و سنسورهای معرفی شده را در محل‌های مورد نظر قرار می‌دهیم (به عنوان مثال رطوبت‌سنج را در خاک گلدان) و سپس به کمک سرورهای طراحی شده در می‌توان به صورت remote از وضعیت گیاه اطلاع یافت.

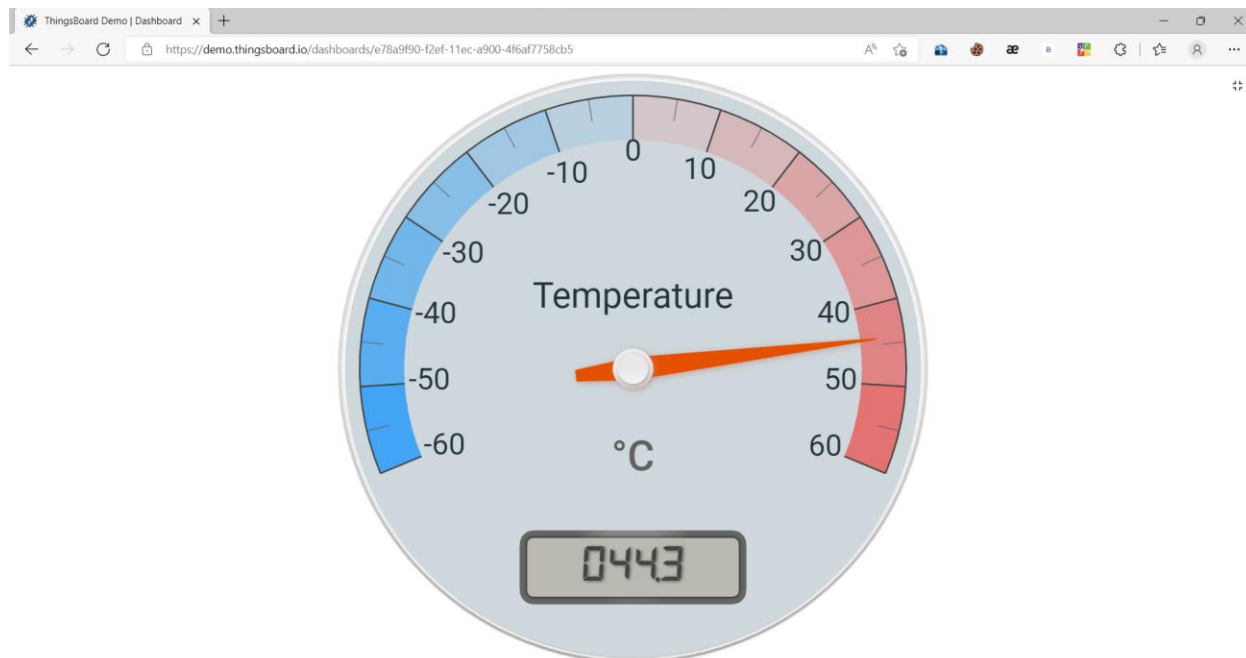


## ۲.۵. صفحات نشان داده شده به کاربر

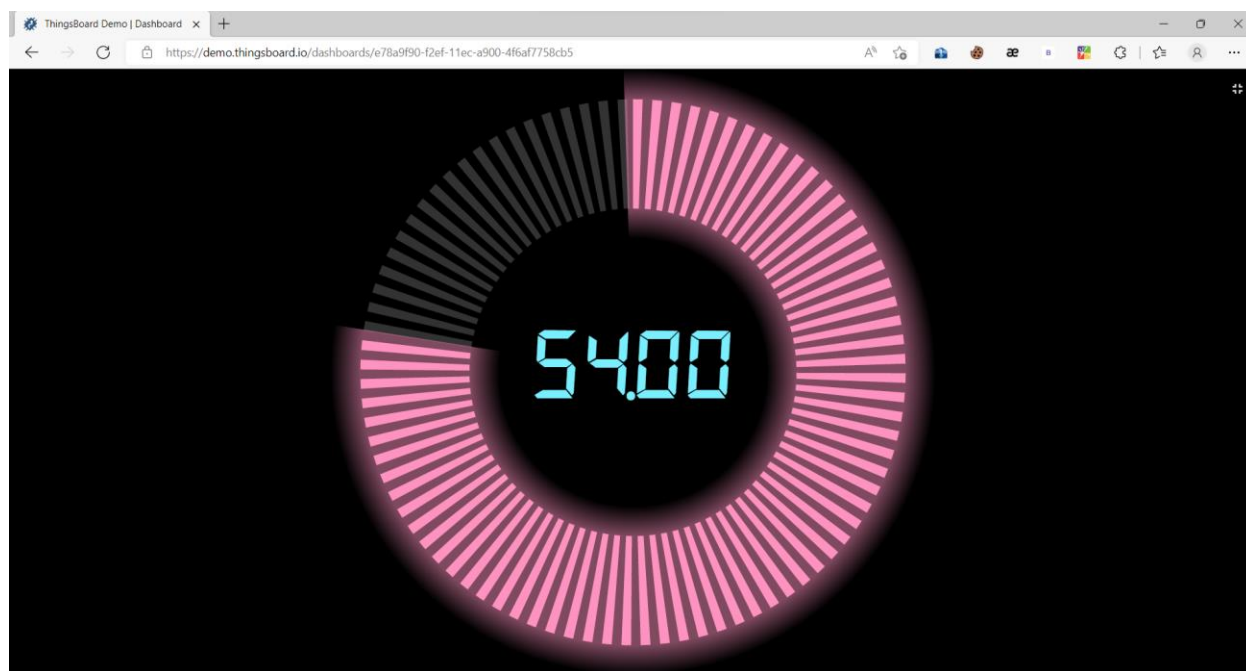
این صفحات در سرور طراحی شده و با اتصال کاربر به آدرس مورد نظر و وارد کردن اطلاعات account امکان دسترسی به آن‌ها فراهم می‌باشد. همانطور که در شکل ۱۸ مشاهده می‌شود، در این داشبورد قابلیت Dark mode نیز موجود می‌باشد که مسپله مهمی برای راحتی کاربران در استفاده از سیستم هنگام نور کم و مکان‌های تاریک می‌باشد. در شکل ۱۹ مشاهده می‌شود که، تغییر نسبت صفحه تأثیری در نمایش Widget ها ندارد و همچنان عملکرد مناسبی دارد. این موضوع امکان استفاده از این سرور روی device های مختلف (تلفن همراه، تبلت و ...) را فراهم می‌سازد.



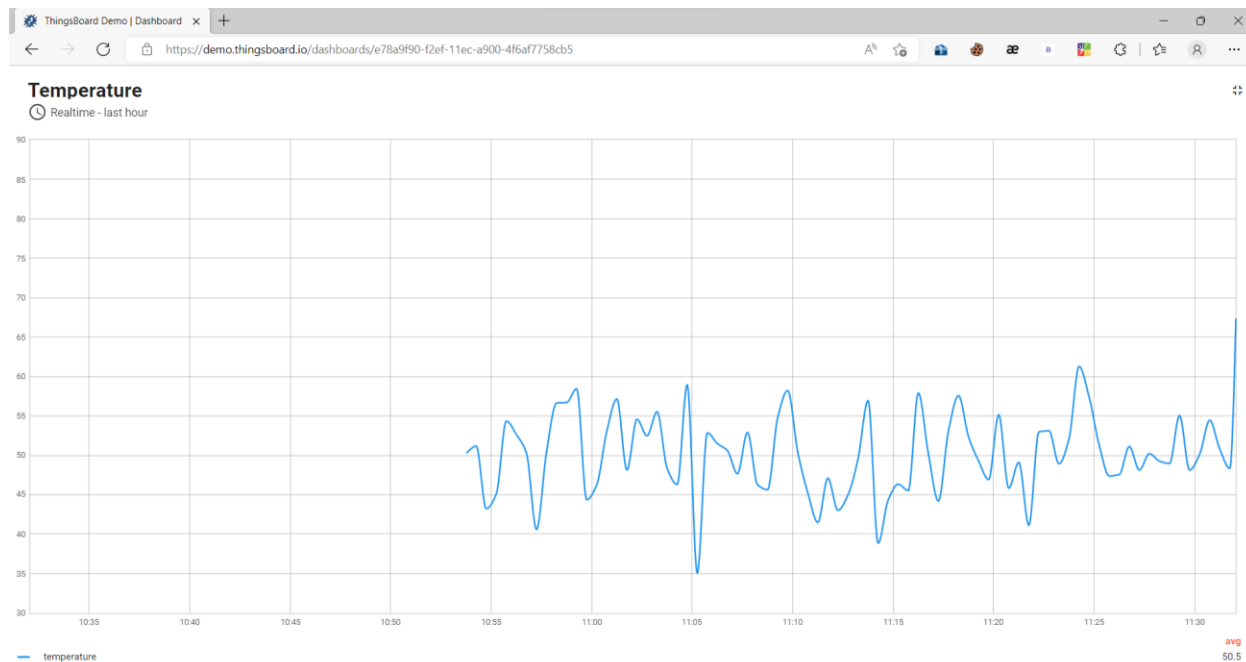
شکل ۱۳: نمای کلی داشبورد نمایش داده شده به کاربر



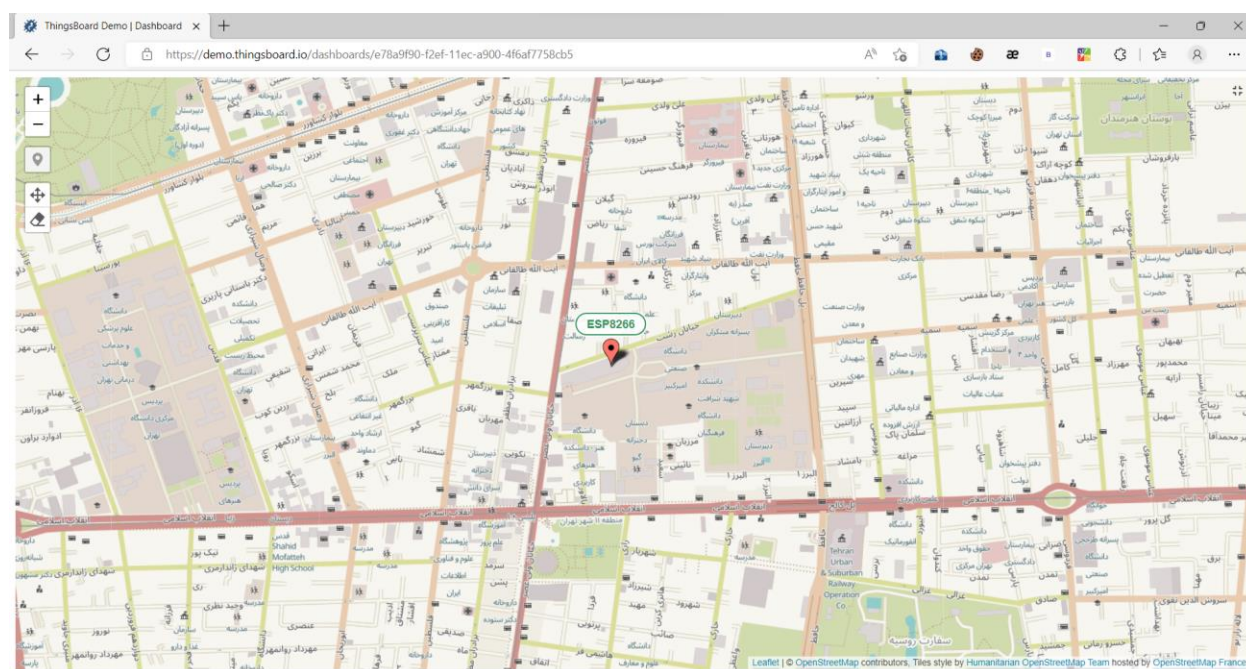
شکل ۱۴: نمای بزرگنمایی شده *Temperature Gauge*



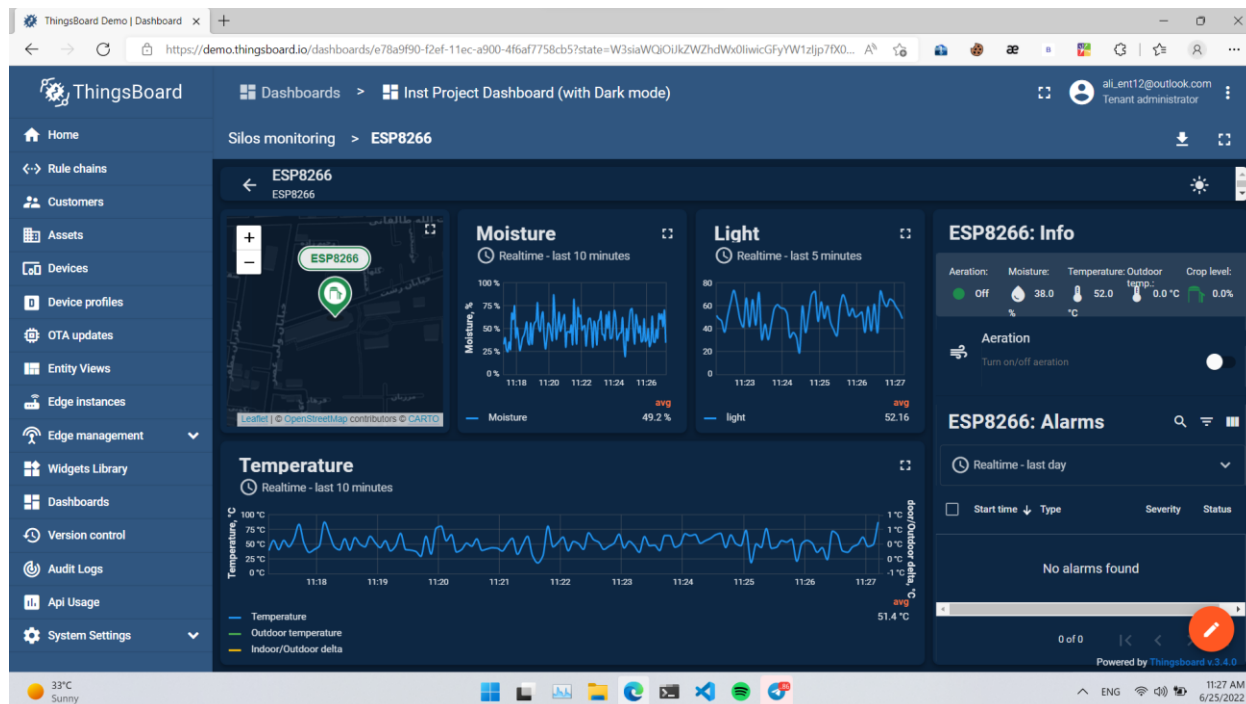
شکل ۱۵: نمای بزرگنمایی شده *Digital Gauge*



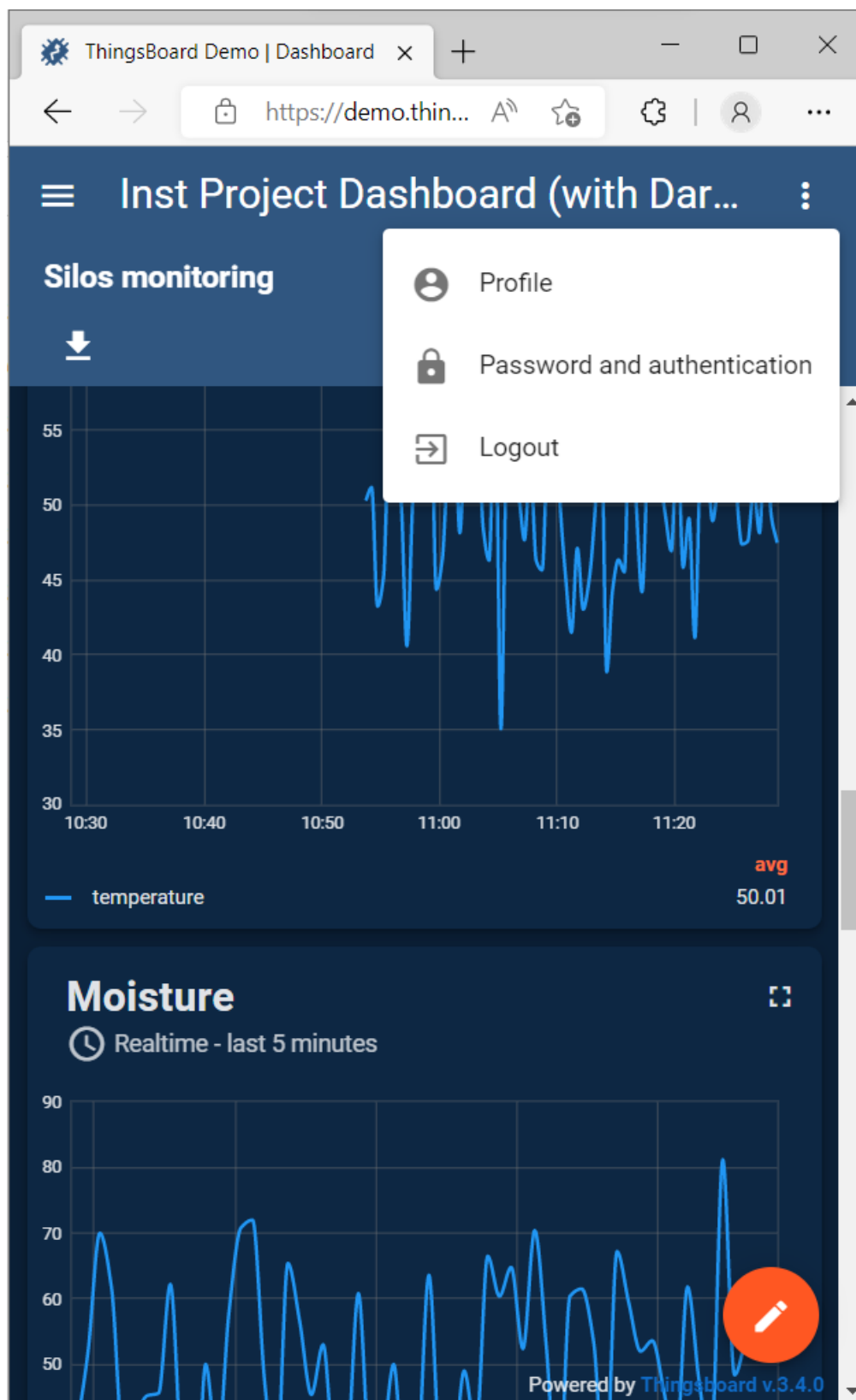
شکل ۱۶: نمای بزرگنمایی شده داده‌ای رکورد شده از Temperature



شکل ۱۷: نقشه موجود در داشبورد برای نمایش موقعیت برد



شکل ۱۸: Dark mode موجود در داشبورد برای راحتی کاربر



شکل ۱۹: استفاده در حالت موبایل

