

REAL-TIME PARTICULATE MATTER MONITORING SYSTEM

فهرست مطالب

صفحه

معرفی بخش‌های مختلف پروژه.....	۳
۱-۱- دستگاه اصلی.....	۳
۲-۱- اپلیکیشن اندروید.....	۵
۳-۱- اپلیکیشن دسکتاپ.....	۷
نرم‌افزار.....	۸
۱-۲- نصب درایور CP2102.....	۸
۲-۲- برنامه‌نویسی ESP32.....	۹
۳-۲- اپلیکیشن موبایل.....	۱۱
۴-۲- اپلیکیشن دسکتاپ.....	۲۲
مدار الکترونیکی.....	۲۷
۱-۳- مدار چاپی.....	۲۷
۲-۳- توضیحات مدار چاپی.....	۲۸
۳-۳- اتصال پایه‌ها.....	۳۳
۴-۳- توضیحات تکمیلی.....	۴۱

معرفی بخش‌های مختلف پروژه

۱-۱- دستگاه اصلی

در شکل زیر، دستگاه اصلی برای نمایش غلظت ذرات نمایش داده شده است. البته در این تصویر، پوست پلکسی گلس کنده نشده است.



در ال سی دی، ۶ مقادیر خروجی سنسور HM3301 که شامل غلظت سه سایز ذره است، نمایش داده میشود. ستون STD مربوط به مقادیر استاندارد $CF=1$ و ستون ATM مربوط به مقادیر اتمسفریک است.

این دستگاه با هر نوع کابل شارژ و شارژر موبایلی روشن میشود و عملکرد کاملی خواهد داشت. همچنین میتوان آن را با کابل USB به کامپیوتر یا لپتاپ وصل کرد، در این حالت هم دستگاه روشن خواهد شد و

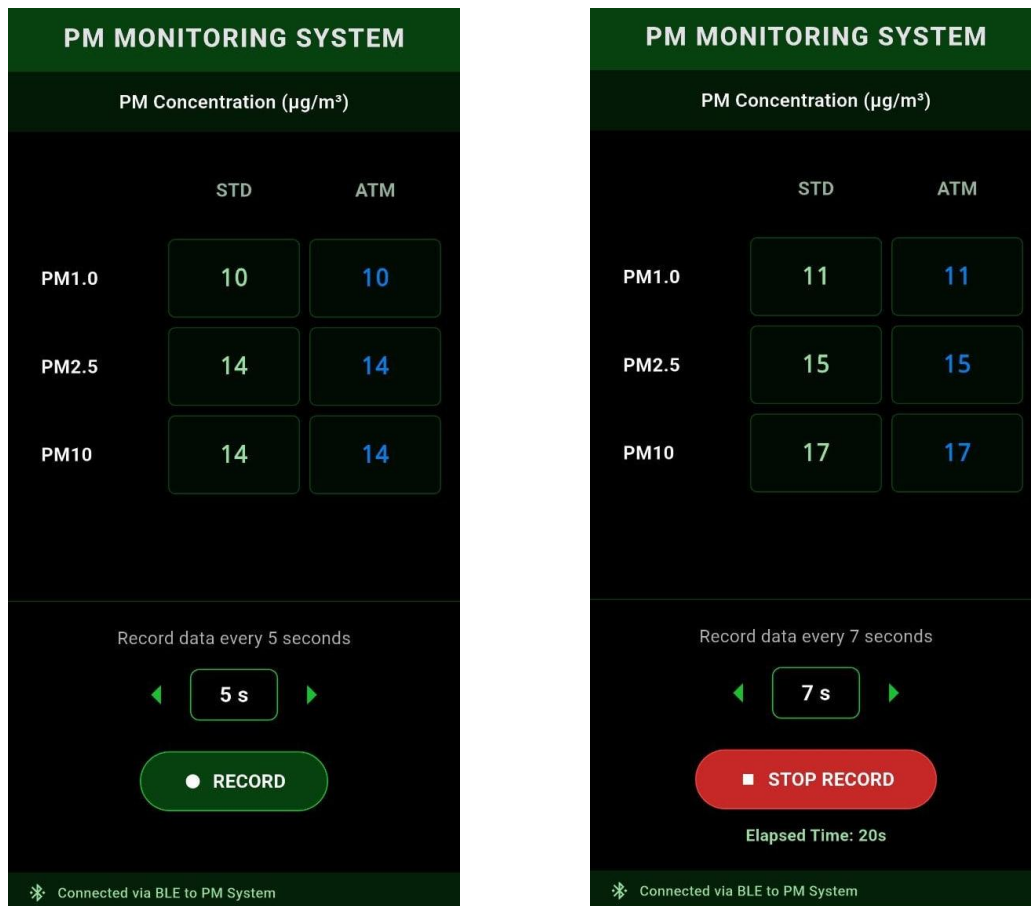
عملکرد کاملی خواهد داشت، و هم با ارتباط سریال کابل USB امکان مشاهده و ثبت داده‌ها و همچنین برنامه نویسی ESP32 وجود خواهد داشت.

نکته بسیار مهم: روی دستگاه، سوکت USB Type-C وجود دارد. برای اتصال به شارژر و کامپیوتر و لپتاپ، میتوان از هر نوع کابلی استفاده کرد، چه کابل دو سر Type-C و چه کابل‌های یک سر Type-C و سر دیگر Type-A. برای اتصال به کامپیوتر، اگر از USB Type-A استفاده مینمایید، حتما از پورت ورژن ۳ به بالا استفاده فرمایید. پورت‌های USB ورژن ۳ دارای پلاستیک ابی رنگ هستند، که جریان بیشتری تامین میکنند. پورت‌های ورژن ۲ دارای پلاستیک سفید هستند و برای تامین جریان دستگاه ایده‌آل نیستند. لطفا از پورت‌های Type-A ورژن ۲ که روی کامپیوتر یا لپتاپ وجود دارد استفاده نفرمایید.

لازم به ذکر است کابل‌های USB دو نوع هستند، برخی صرفا برای شارژ استفاده میشوند و سیم برای دیتا ندارند. اگر از این کابل‌ها برای اتصال به کامپیوتر استفاده شود، دستگاه فقط روشن میشود و امکان ارتباط با دستگاه وجود نخواهد داشت.

۱-۲- اپلیکیشن اندروید

در تصویر زیر، محیط اپلیکیشن اندروید نمایش داده شده است. این اپلیکیشن از طریق بلوتوث BLE با دستگاه اصلی ارتباط برقرار میکند.



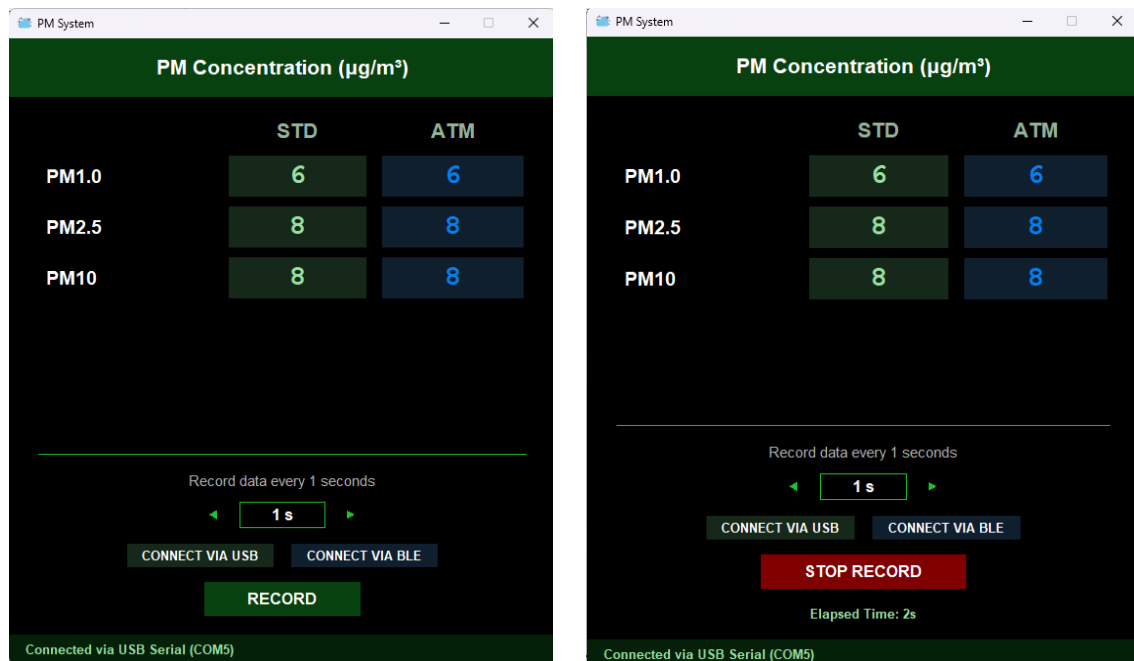
بلوتوث BLE نیاز به دسترسی Location دارد. پس از نصب اپلیکیشن، در بخش Permissions دسترسی Location را به این اپلیکیشن داده، سپس کافی است بلوتوث و لوکیشن گوشی را روشن نمایید، برنامه به صورت خودکار اسکن کرده و به دستگاه اصلی متصل میشود.

نکته مهم: از طریق بخش بلوتوث گوشی نباشد دستگاه pair شود. نیاز به انجام هیچ کاری نیست، فقط باید بلوتوث گوشی فعال شود. برنامه به صورت خودکار اتصال را برقرار میکند.

در این اپلیکیشن، علاوه بر دیدن نتایج، میتوانید داده‌ها را با فاصله زمانی مورد نظر خود ذخیره نمایید. داده‌ها در فایل txt و در مسیر Internal Storage > Documents ذخیره میشوند. در این فایل، داده‌ها با Tab از یکدیگر جدا شده‌اند که این امر به ایپورت ساده‌تر داده‌ها در متلب و اکسل و ... کمک میکند. لازم به ذکر است در این پیکربندی، بلوتوث BLE به صورت یک به یک کار میکند، یعنی اگر با یک گوشی به آن متصل شوید، امکان اتصال دستگاه دیگری وجود نخواهد داشت، تا زمانی که دستگاه قبلی قطع شود.

۱-۳- اپلیکیشن دسکتاپ

اپلیکیشن دسکتاپ در تصویر زیر نمایش داده شده است. این اپلیکیشن، همان قابلیت‌های اپلیکیشن موبایل را خواهد داشت، با این تفاوت که علاوه بر بلوتوث BLE، با استفاده از کابل USB نیز امکان اتصال وجود دارد. پس از اجرای برنامه، به دلخواه خود میتوانید روش اتصال را انتخاب کرده تا به دستگاه وصل شوید.



نکته مهم: برای اتصال بلوتوث، تنها کافی است بلوتوث کامپیوتر روشن باشد. اصلاً نباید از طریق ویندوز دستگاه شناسایی و pair شود. هیچ کاری لازم نیست انجام شود جز زدن دکمه CONNECT VIA BLE. با زدن این دکمه، به سرعت اسکن شروع شده و ارتباط برقرار میگردد.

در این اپلیکیشن نیز میتوانید با فاصله زمانی مطلوب داده‌ها را ذخیره نمایید. فایل داده‌ها در دسکتاپ شما ایجاد میشود.

۲-۱- نصب درایور CP2102

برد ESP32 مورد استفاده، از مبدل CP2102 استفاده می کند، برای ارتباط آن با کامپیوتر، لازم است درایور CP2102 روی کامپیوتر نصب شود. برای دانلود درایور، لطفاً به آدرس زیر مراجعه فرمایید و فایل CP210x Universal Windows Driver را از تب Downloads دانلود نمایید (فایل دانلود شده و لینک آهن در فولدر Resources قرار دارد).

<https://www.silabs.com/software-and-tools/usb-to-uart-bridge-vcp-drivers?tab=downloads>

فایل را Extract کرده، سپس روی فایل silabser.inf راست کلیک و گزینه Install را انتخاب نمایید. در صورت نصب موفقیت آمیز، پنجره‌ای با همین پیام باز می شود. حالا اگر برد ESP32 را به کامپیوتر متصل کرده به Device Manager مراجعه نمایید، در بخش Ports (COM & LPT) برد ESP32 با عنوان Silicon Labs CP210x USB to UART Bridge (COM5) اضافه شده است.

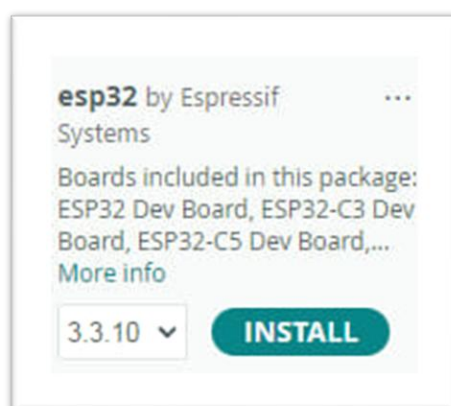


۲-۲- برنامه‌نویسی ESP32

برای برنامه‌نویسی ESP32 می‌توان از نرم‌افزار اوپن سورس Arduino IDE استفاده کرد که برای برنامه‌نویسی خود بردهای آردوینو نیز استفاده می‌شود. با توجه به سیستم عامل، نسخه مناسب دانلود و نصب می‌گردد (با اینترنت ایران نصب نمی‌شود). لینک دانلود نرم‌افزار Arduino IDE :

<https://www.arduino.cc/en/software/>

بعد از نصب نرم‌افزار، از منوی بالا وارد Tools و سپس Board و در نهایت Boards Manager شوید. یک تب سمت چپ برنامه باز می‌شود. در این لیست، ESP32 (By Espressif Systems) را نصب نمایید. (با اینترنت ایران نصب نمی‌شود).



پس از نصب ESP32، نرم‌افزار Arduino IDE بردهای ESP32 را خواهد شناخت. برد را از طریق کابل USB به کامپیوتر وصل نمایید، سپس از منوی Tools برد زیر را انتخاب نمایید:

Tools > Board > esp32 > ESP32 Dev Module

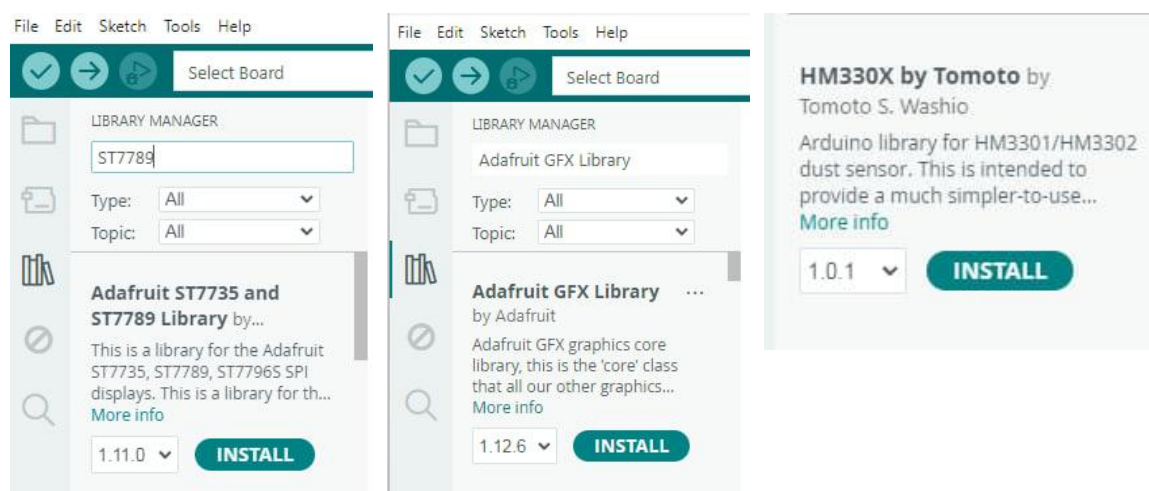
سپس از مسیر زیر، پورت ESP32 را انتخاب نمایید:

Tools > Port > COMx

در شرایط مختلف، پورت می‌تواند متفاوت باشد. مثلاً COM3 یا COM5 و برای یافتن پورت دقیق، می‌توانید از Device Manager استفاده نمایید. مطابق عکس زیر، پورت من COM5 است.



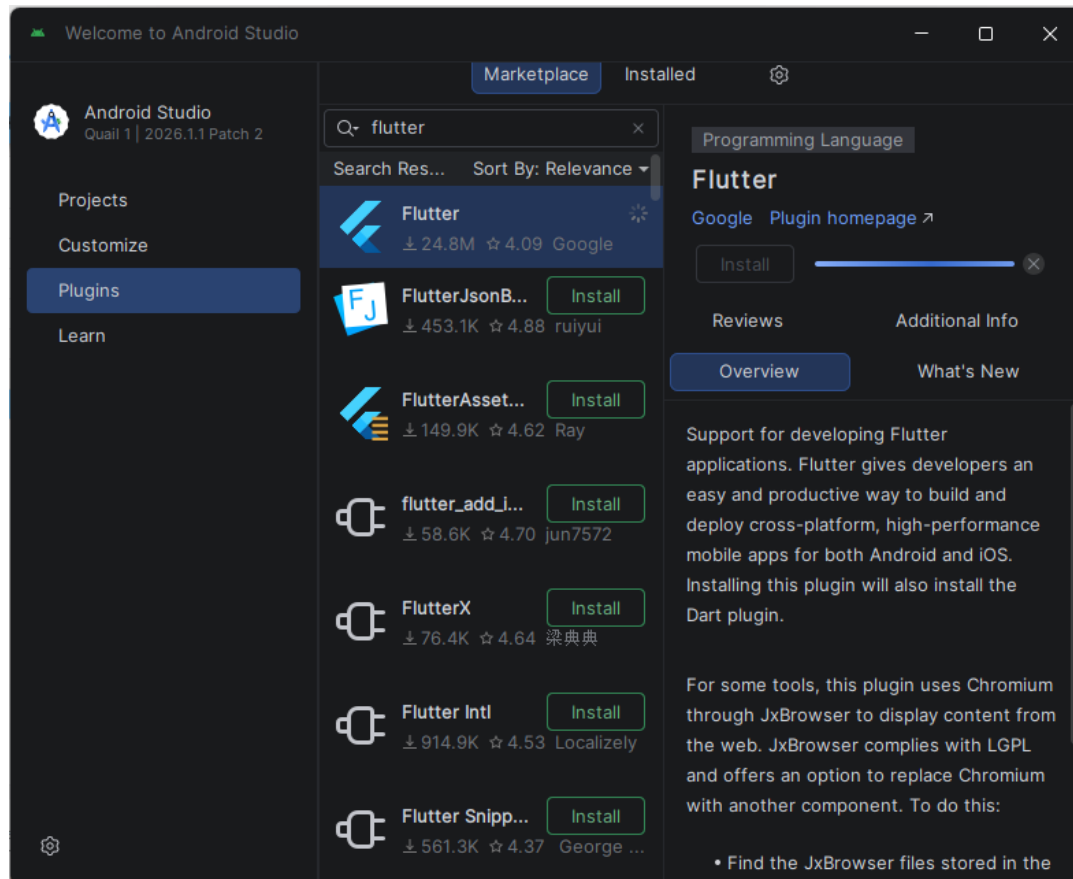
اکنون اتصال برد به کامپیوتر کامل شده است و امکان آپلود کد روی آن وجود دارد. کدی که برای ESP32 نوشته شده است، دارای چند کتابخانه است که آنها نیز باید نصب شوند. هنگام نصب، پرسیده می‌شود که وابستگی‌ها نیز نصب شوند یا خیر، که باید تمام وابستگی‌ها نیز نصب شود.



با نصب این کتابخانه‌ها، با زدن دکمه Upload، کد روی ESP32 آپلود شده و دستگاه شروع به کار خواهد کرد. کدهای ESP32 در فولدر ESP32 قرار دارد.

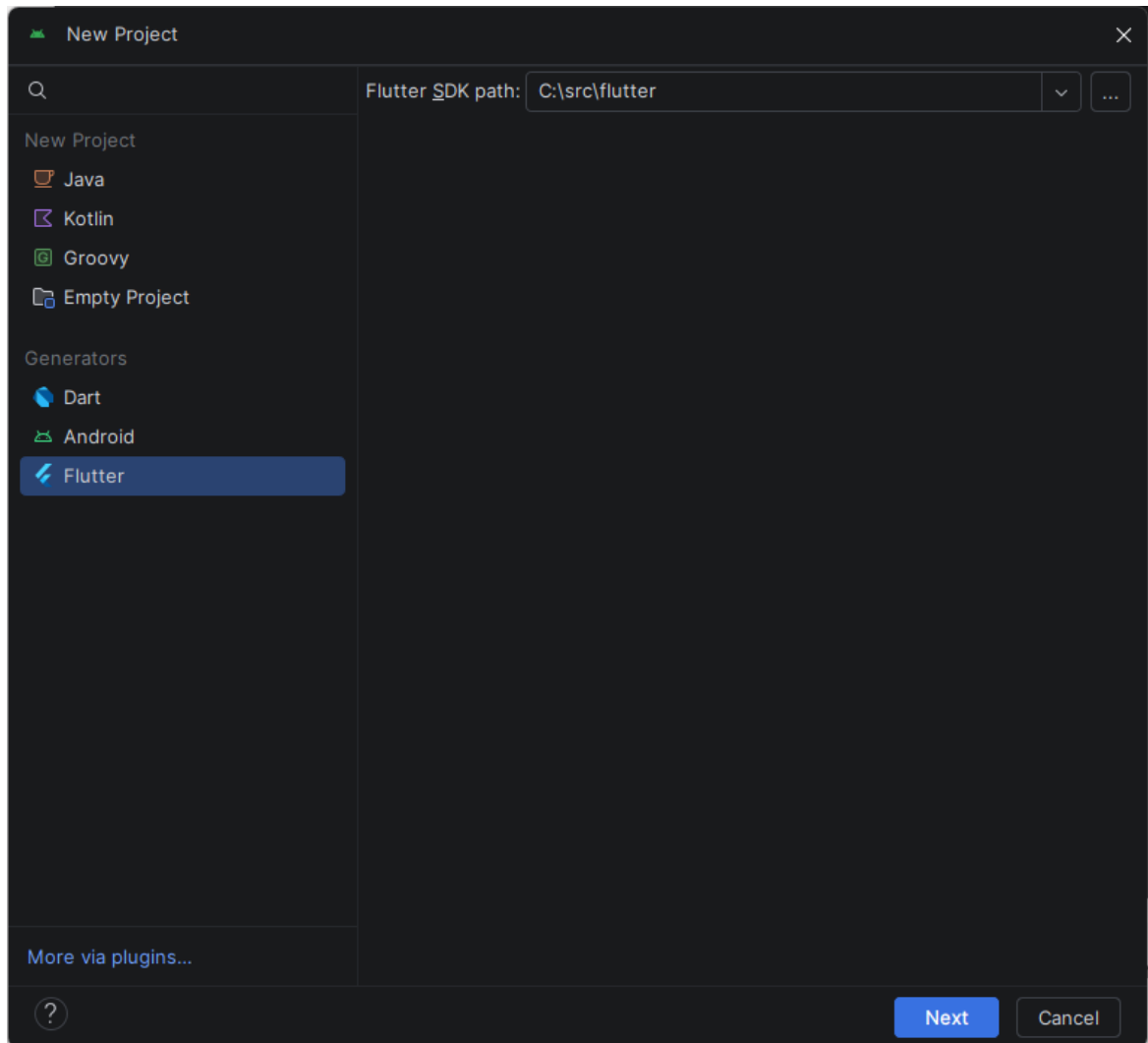
۲-۳- اپلیکیشن موبایل

پس از نصب آخرین نسخه نرم افزار Android Studio، که رایگان است، از منوی سمت چپ گزینه Plugins انتخاب شده و در آن کلمه Flutter سرچ شده و پلاگین فلاتر نصب میگردد.



سپس باید Flutter نصب شود، که میتوان از سایت رسمی آن دانلود و سپس در یک مسیر مشخص نصب کرد. من در مسیر `C:\src\flutter` نصب کرده ام.

حال مجدداً Android Studio را باز کرده، در صفحه اصلی گزینه New Flutter Project را انتخاب میکنیم. در متوی سمت چپ، حتماً گزینه Flutter باید انتخاب شود. سپس باید Flutter SDK Path را مشخص کرد، همان جایی است که فلاتر نصب شده. برای من میشود: `C:\src\flutter`



سپس به این صورت تنظیم میکنیم:

New Project

Project name:

Project location: ...

Description:

Project type:

Organization:

Android language: ☐ Java ☒ Kotlin

Platforms: ☒ Android ☐ iOS ☐ Linux ☐ MacOS ☐ Web ☐ Windows

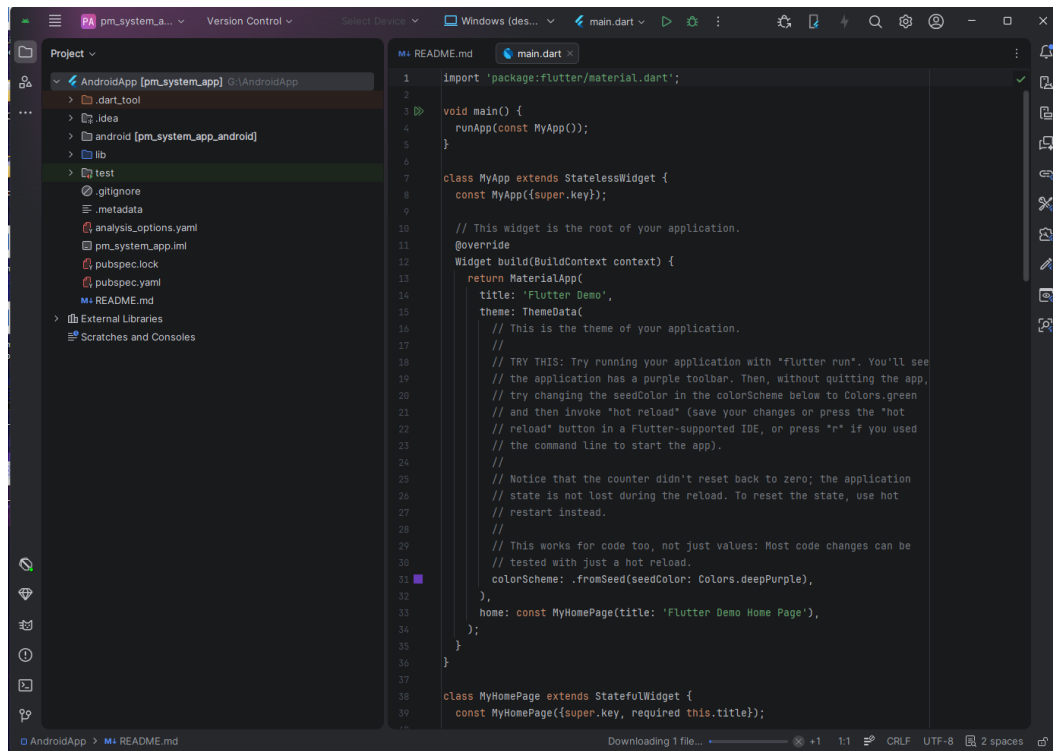
When created, the new project will run on the selected platforms (others can be added later).

☐ Create project offline

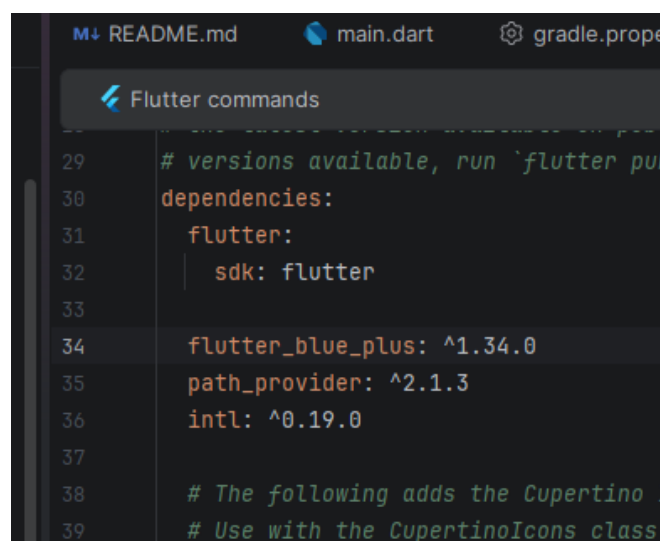
> More Settings

? Previous Create Cancel

در نهایت صفحه ای به صورت زیر باز میشود:



در منوی سمت چپ، فایل pubspec.yaml را یافته و دوبار کلیک میکنیم تا باز شود. سپس در این فایل، وابستگی های مورد نیاز را در جای مناسب اضافه میکنیم، مانند تصویر زیر:



در این فایل، سه خط زیر اضافه شده اند که باید دقیقاً تراز با flutter باشند.

```
flutter_blue_plus: ^1.34.0
```

```
path_provider: ^2.1.3
```

```
intl: ^0.19.0
```

پس از اضافه شدن این خطوط، در نوار بالا، گزینه Pub get زده میشود.

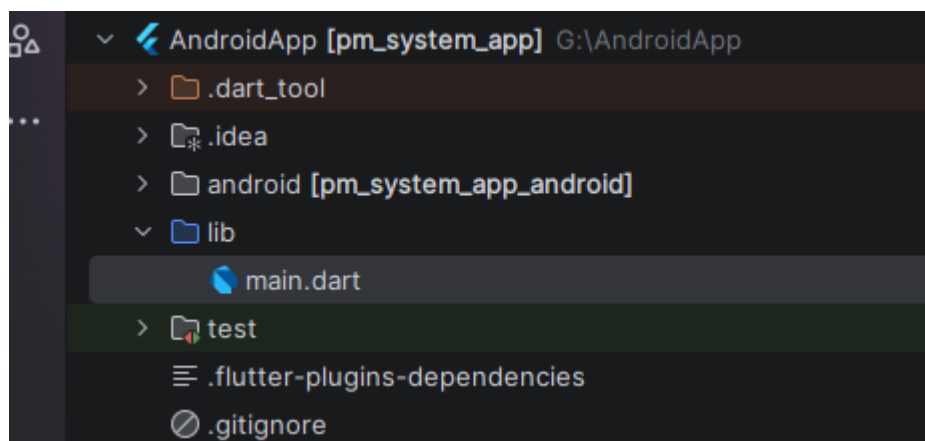
```
er commands
Pub get Pub upgrade Pub ou

versions available, run 'flutter pub outdated'.
dependencies:
flutter:
  sdk: flutter

flutter_blue_plus: ^1.34.0
path_provider: ^2.1.3
intl: ^0.19.0

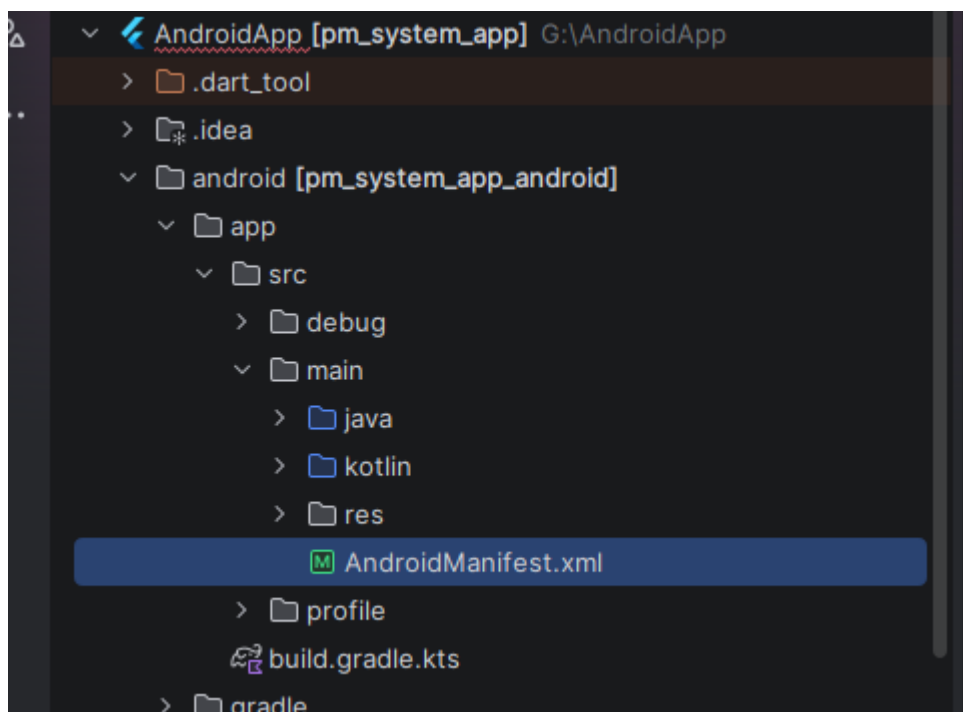
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
```

وقتی انجام شد، از منوی سمت چپ، لیست lib را باز کرده و گزینه main.dart را با دوبار کلیک انتخاب میکنیم.



کل کدهای پیش فرض را حذف و کدهای اصلی اپلیکیشن را در آن مینویسیم. برای ذخیره هر تغییر، میتوان به سادگی از `ctrl + s` استفاده نمود.

اکنون از منوی سمت چپ، فایل AndroidManifest.xml را انتخاب میکنیم:



دقیقا زیر تگ manifest و قبل از تگ application کد زیر را قرار میدهیم.

```
<uses-permission android:name="android.permission.BLUETOOTH" />
```

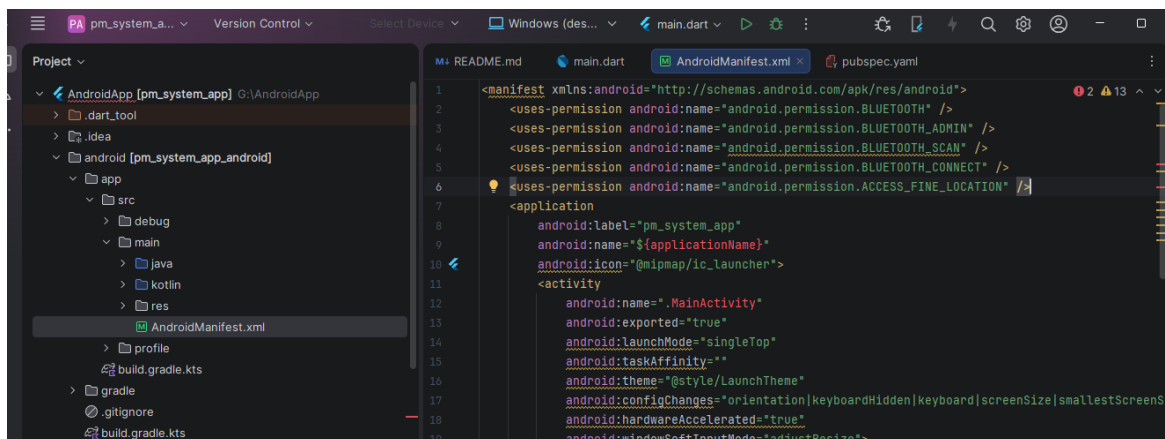
```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

```
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
```

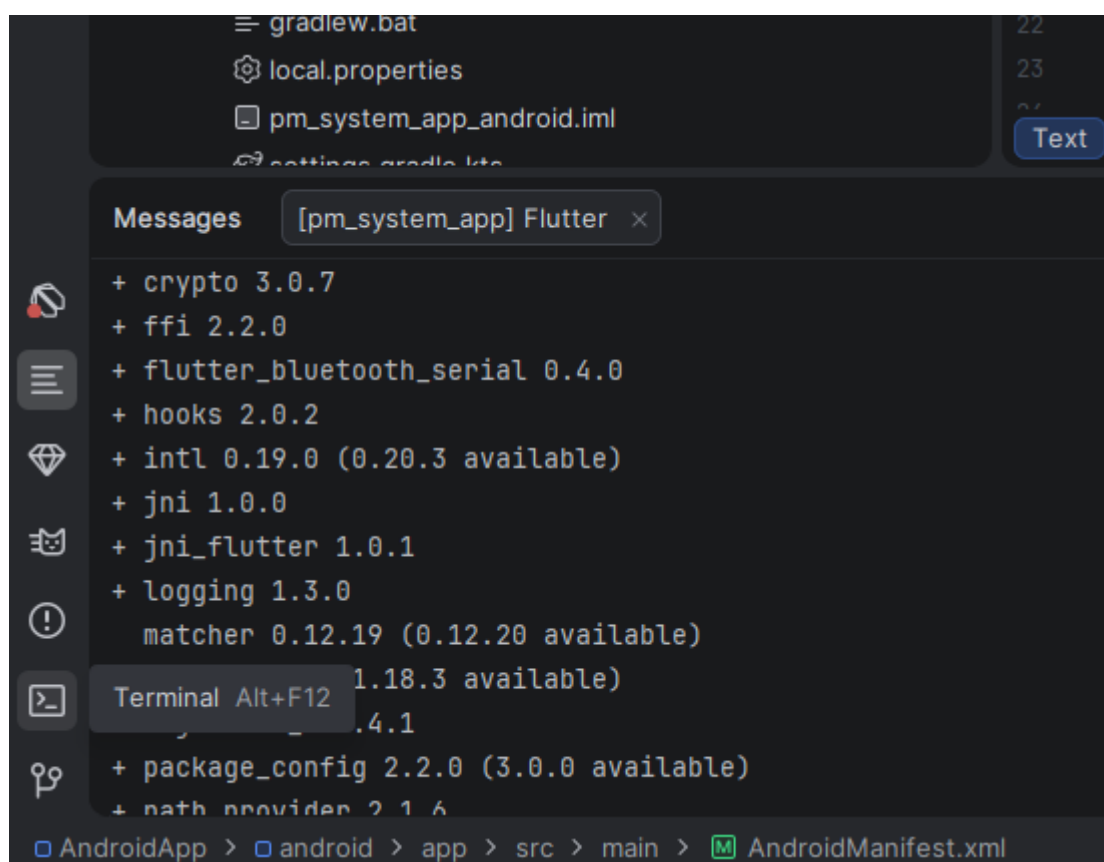
```
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
```

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```


به این صورت میشود: (ذخیره فراموش نشود)

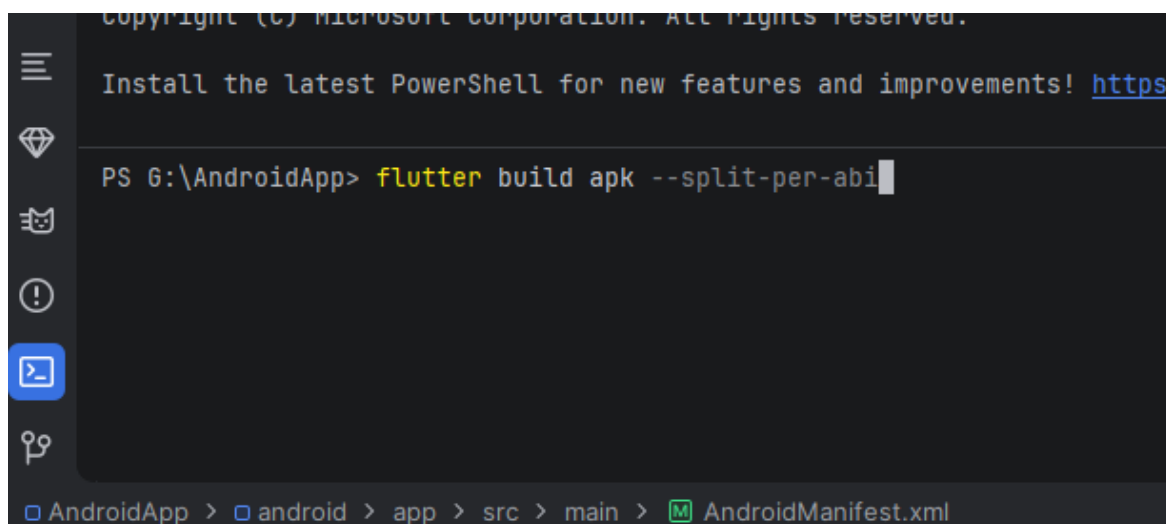


برای تولید فایل نصب نهایی با فرمت APK از نوار کنار گزینه Terminal را انتخاب میکنیم.



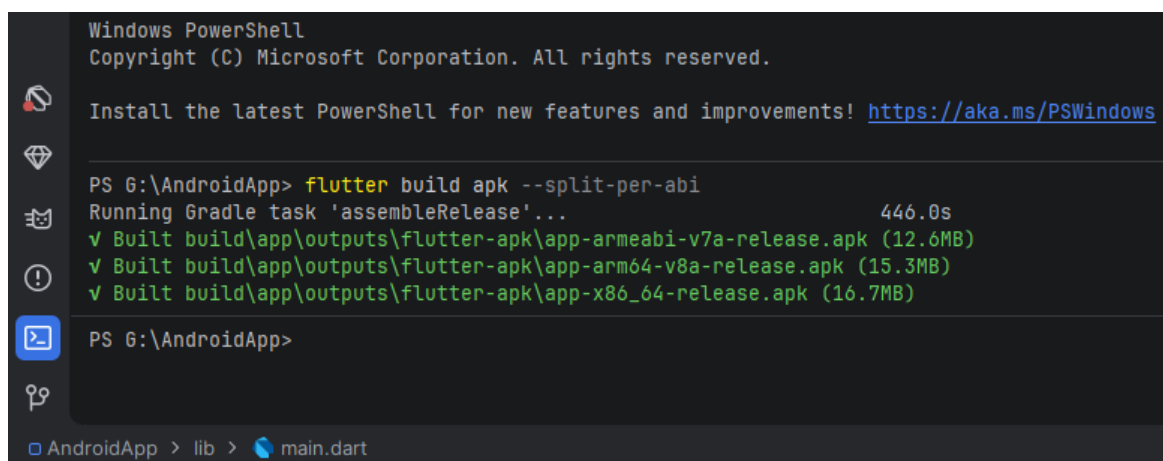
دستور زیر را در ترمینال تایپ کرده و اینتر میزنیم:

```
flutter build apk --split-per-abi
```



```
Copyright (C) Microsoft Corporation. All rights reserved.  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS G:\AndroidApp> flutter build apk --split-per-abi
```

پس از زدن اینتر، صبر میکنیم تا فایلها ساخته شود. به این صورت:



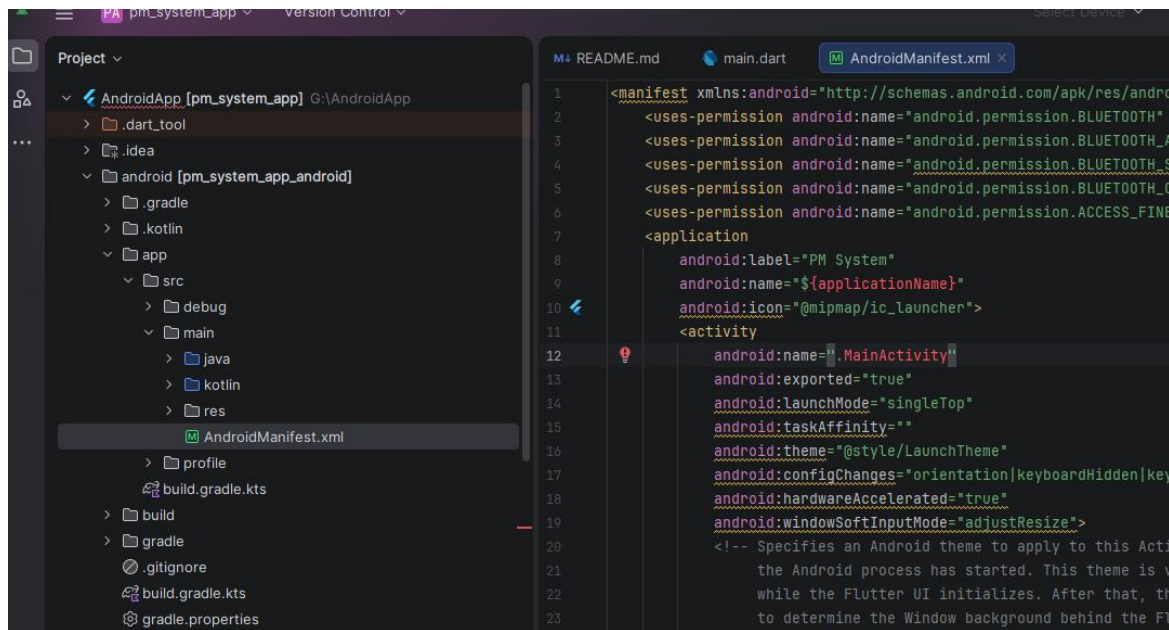
```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS G:\AndroidApp> flutter build apk --split-per-abi  
Running Gradle task 'assembleRelease'... 446.0s  
✓ Built build\app\outputs\flutter-apk\app-armeabi-v7a-release.apk (12.6MB)  
✓ Built build\app\outputs\flutter-apk\app-arm64-v8a-release.apk (15.3MB)  
✓ Built build\app\outputs\flutter-apk\app-x86_64-release.apk (16.7MB)  
  
PS G:\AndroidApp>
```

پس از اتمام، وارد مسیری که پروژه را در آن ساخته شده است میشویم که برای من `G:\AndroidApp` بود، سپس در آن فولدر، به مسیر `build\app\outputs\flutter-apk` رفته و فایل نصب `app-arm64-v8a-release.apk` را استفاده میکنیم. این فایل برای اکثر گوشی های مدرن قابل نصب است.

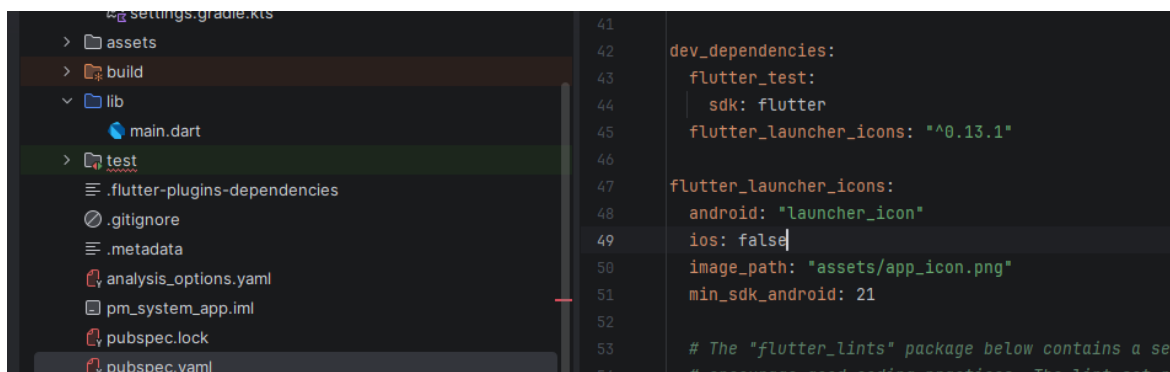
نکته ۱: برای تغییر نام اپلیکیشن، به مسیر

```
android > app > src > main > AndroidManifest.xml
```

رفته و نام دلخواه خود را در این خط جایگذاری میکنیم: `android:label="PM System"`



نکته ۲: تعیین آیکون برای اپلیکیشن. یک آیکون مناسب انتخاب میکنیم، با سایز مثلا ۵۱۲ در ۵۱۲ پیکسل، و سپس یک فولدر در مسیر ایجاد پروژه (که برای من G:\AndroidApp بود) به نام assets ایجاد میکنیم و تصویر آیکون را مثلا با نام app_icon.png در آن قرار میدهیم. فرمت png انتخاب مناسبی است. مجدداً به فایل pubspec.yaml برمیگردیم و بخش dev_dependencies را به صورت زیر ویرایش میکنیم و پکیج flutter_launcher_icons را اضافه میکنیم. پس از آن باید Pub get در نوار بالا زده شود.



در فیلد `image_path` باید مسیر نسبی فایل آیکون نوشته شود که برای من به این صورت بود. در ضمن به فاصله کدهای نوشته شده از دیواره سمت چپ دقت شود، باید مرتب و درست باشند. حتما `ios` را `flase` نمایید.

سپس در ترمینال، کد زیر را اجرا نمایید:

```
flutter pub run flutter_launcher_icons
```

سپس قبل از ساخت فایل APK، کد زیر را در ترمینال اجرا میکنیم؛ با این دستور کش و فایل‌هایی که قبلا ساخته شده اند حذف میشوند:

```
flutter clean
```

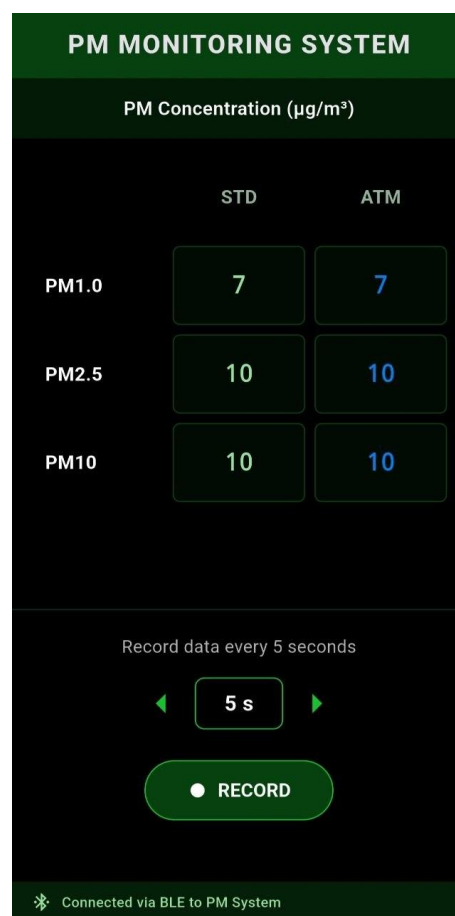
و در نهایت فایل نصبی را میسازیم.

نکته بسیار بسیار مهم ۱: با توجه به اینکه از بلوتوث BLE استفاده شده است، برای اتصال اپلیکشن به برد ESP32 نباید از قسمت بلوتوث گوشی، برد ESP32 که اینجا نام آن را PM System است، اتصال برقرار شود (Pair شود)، چون در این حالت، گوش یاز حالت کلاسیک ارتباط را برقرار کرده و دیگر اپلیکشن امکان برقراری ارتباط نخواهد داشت. بنابراین باید گذاشت تا اپلیکشن خودش اسکن کرده و ارتباط را برقرار نماید.

نکته بسیار بسیار مهم ۲: بلوتوث BLE نیاز دارد به مجوز Location دارد که باید از مسیر زیر این مجوز به اپلیکیشن داده شود:

Settings > Apps > PM System > Permissions > location

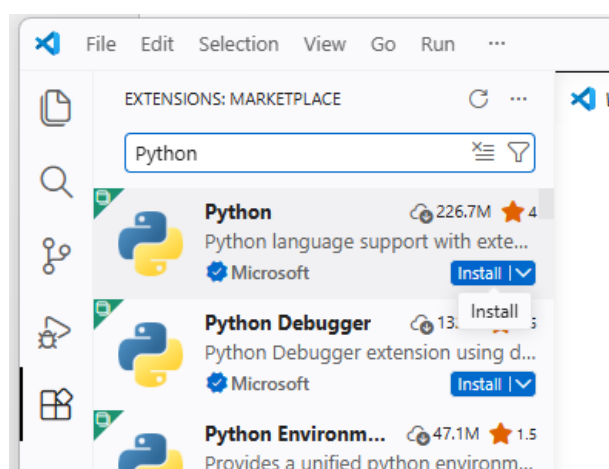
سپس باید Location گوشی روشن شود. پس از انجام این نکات، اپلیکیشن به درستی کار خواهد کرد.
نکته: فایل txt دیتاهای ذخیره شده، در مسیر Internal Storage > Documents قرار دارد و نام فایل بر اساس تاریخ و زمان ساخته میشود. در شکل زیر، تصویر اپلیکیشن اندروید نمایش داده شده است.



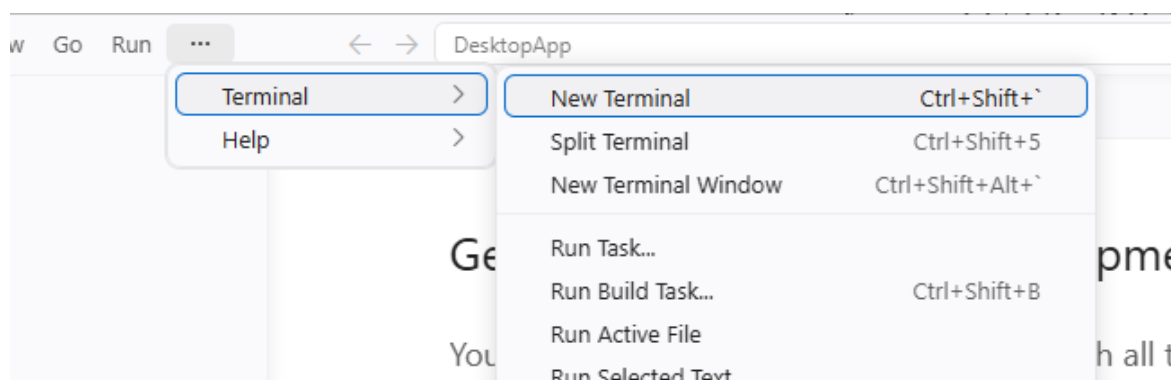
۲-۴- اپلیکیشن دسکتاپ

برای ساخت اپلیکیشن دسکتاپ، از زبان پایتون بهره میبریم. ابتدا آخرین نسخه پایتون را از سایت رسمی آن گرفته و نصب میکنیم. هنگام نصب، حتما گزینه Add python.exe to PATH انتخاب شود تا در ترمینال ویندوز دستورات قابل شناسایی باشد.

برای کدنویسی پایتون، از نرم افزار VS Code استفاده میشود. این نرم افزار نیز دانلود و نصب میگردد. پس از باز کردن نرم افزار، در نوار سمت چپ روی آیکون Extensions کلیک کرده و Python را سرچ کرده و آن را نصب میکنیم.



برای پروژه یک فولدر مثلا با نام DesktopApp میسازیم. سپس از منوی بالا، گزینه File و سپس Open Folder را انتخاب کرده و فولدر ساخته شده را انتخاب میکنیم. از منوی بالا، گزینه Terminal و سپس New Terminal را انتخاب میکنیم، مثل تصویر زیر:

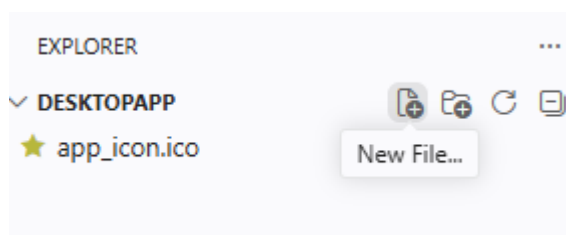


سپس دستور زیر را برای نصب کتابخانه‌های لازم اجرا میکنیم:

```
pip install pyserial bleak
```

همچنین برای آیکون اپلیکیشن دسکتاپ، باید فرمت آیکون خود را ico کنیم. میتوان از کانورترهای آنلاین برای این کار استفاده کرد. سپس فایل آیکون را که در اینجا app_icon.ico است را در همان فولدر DesktopApp قرار میدهیم. وقتی فایل آیکون در فولدر قرار گیرد، در VS Code در منوی Explorer دیده خواهد شد.

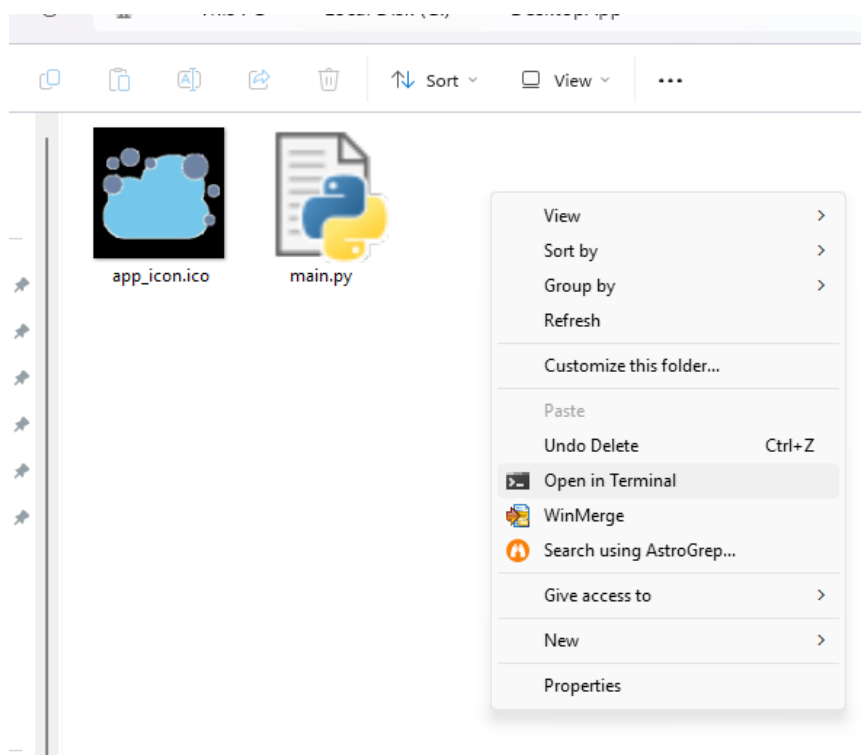
برای نوشتن کدهای پایتون، در منوی سمت چپ که Explorer است، روی آیکون New File کلیک میکنیم. و نام آن را main.py میگذاریم.



سپس کدهای لازم را نوشته و ذخیره میکنیم. برای تولید فایل نصبی exe. که امکان نصب روی کامپیوتر و لپتاپ را داشته باشد و نیازی هم به پایتون و کتابخانه و .. نداشته باشد، در ترمینال دستور زیر را اجرا میکنیم:

```
pip install pyinstaller
```

حال وارد فولدر پروژه میشویم، جایی که فایل `main.py` و فایل آیکون وجود دارد. سپس کلیک راست کرده و ترمینال را در همان فولدر باز میکنیم.

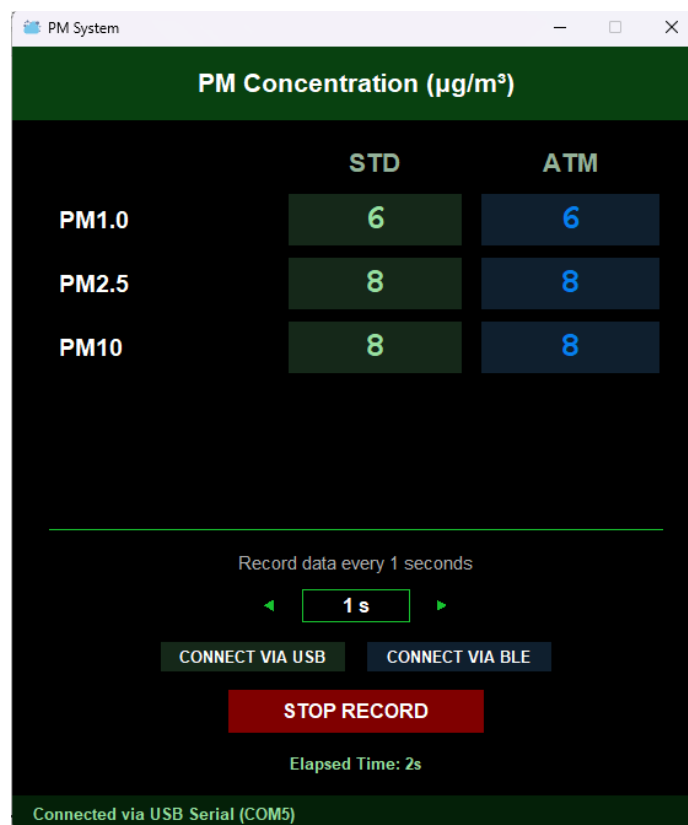
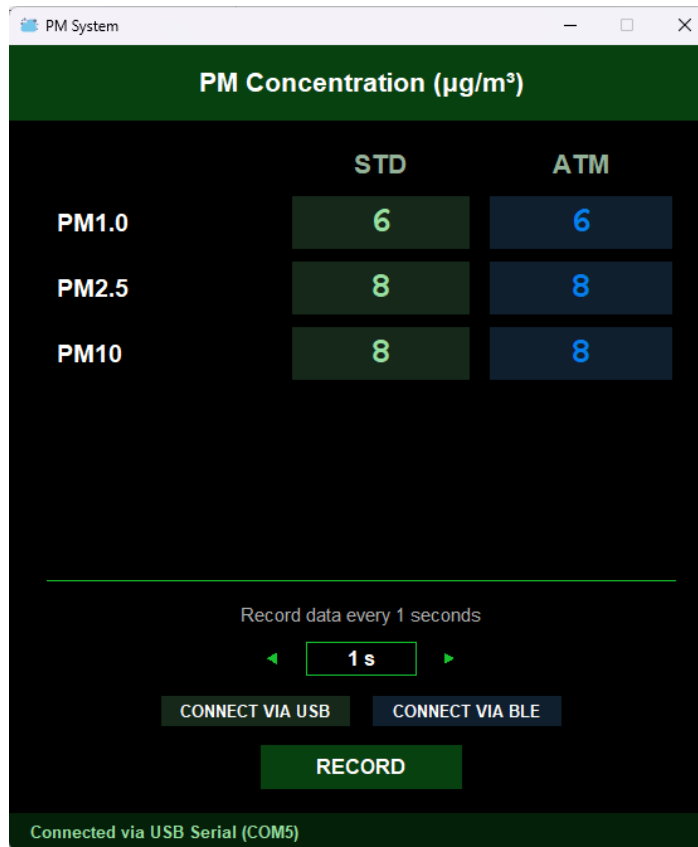


در ترمینال، دستور زیر را اجرا میکنیم:

```
pyinstaller --noconsole --onefile --icon=app_icon.ico --add-data "app_icon.ico;" main.py
```

منتظر میمانیم تا کارش تمام شود. پس از اتمام، چند فولدر در همانجا ساخته میشود. فایل نهایی در فولدر `dist` قرار دارد. در شکل زیر، تصویر اپلیکیشن دسکتاپ نمایش داده شده است. در این اپلیکیشن، به دلخواه، میتوان از طریق کابل USB یا بلوتوث به دستگاه متصل شد. تنها کافیست روی دکمه مورد نظر کلیک شود.

همچنین برای ذخیره دیتاها، میتوان زمان بالا را به دلخواه تنظیم کرد، مثلاً هر ۱ ثانیه دیتاها ذخیره شود. دیتاها در فایل `txt` به همراه تاریخ و ساعت در دسکتاپ ذخیره میشود. ساختار ذخیره دیتاها در فایل `txt` به گونه‌ای است که با `tab` جدا شده اند تا به سادگی بتوان آن را به متلب یا اکسل منتقل کرد.



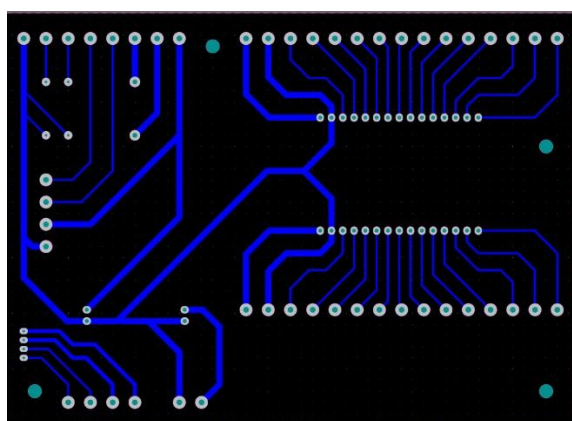
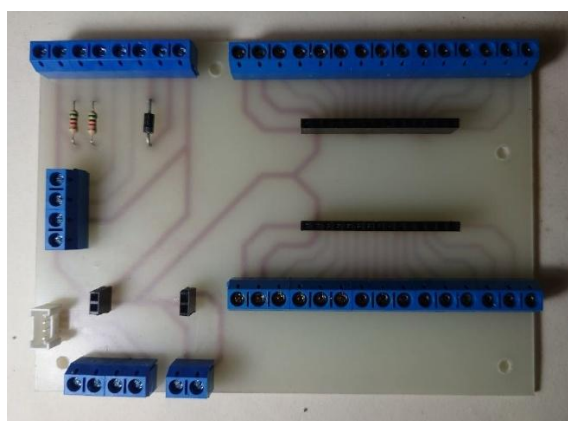
نکته بسیار مهم: برای اینکه کامپیوتر از طریق کابل USB برد ESP32 را بشناسد، لازم است درایور CP2102 که در شکسن ۱-۱ شرح داده شد، روی کامپیوتر نصب باشد.

نکته بسیار مهم: همانند اپلیکیشن موبایل، در اپلیکیشن دسکتاپ نیز نباید از طریق ویندوز و تنظیمات، بلوتوث pair شود. باید اجازه داد تا اپلیکیشن خودش اسکن کرده و متصل شود به بلوتوث. در واقع برای اتصال بلوتوث، هیچ کاری نباید انجام شود. در مورد موبایل هم به همین شکل است، خود برنامه همه کار را انجام میدهد.

مدار الکترونیکی

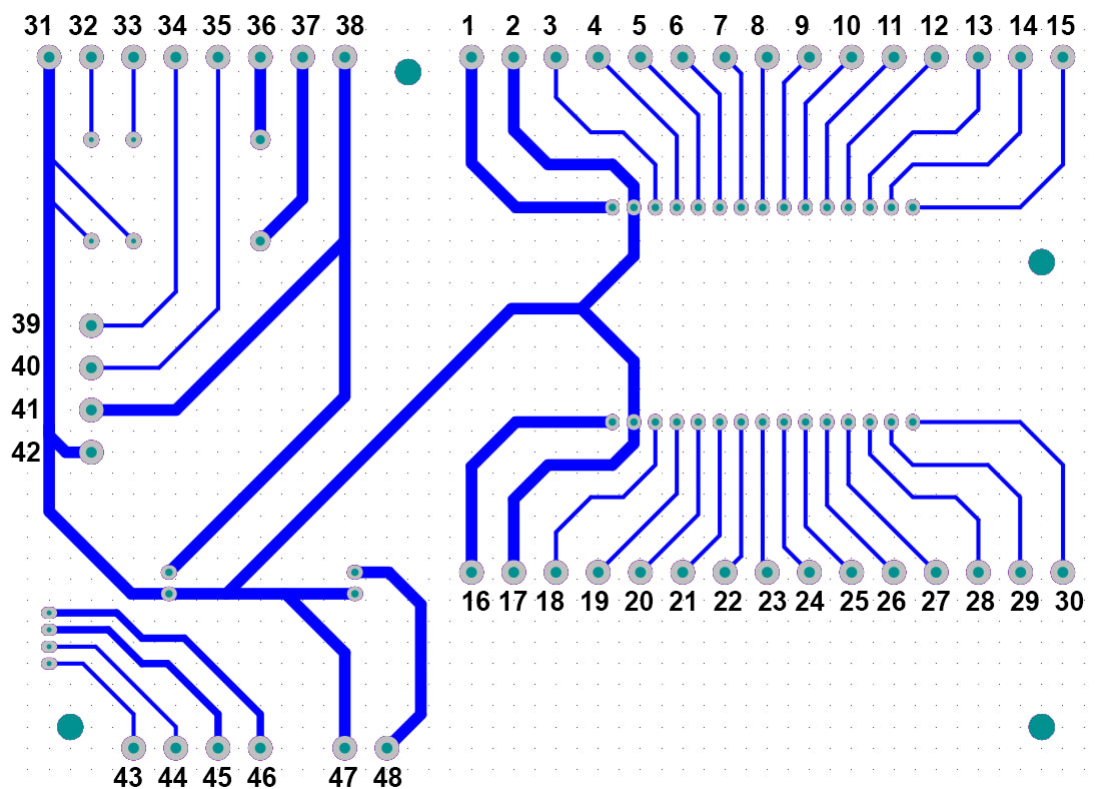
۳-۱- مدار چاپی

برای کاربری ساده‌تر و اتصالات محکم و قابل اعتماد، مدار چاپی به صورت زیر طراحی و ساخته شده است. نقشه این مدار که با استفاده از نرم افزار Altium طراحی شده است، در فولدر PCB قرار دارد.

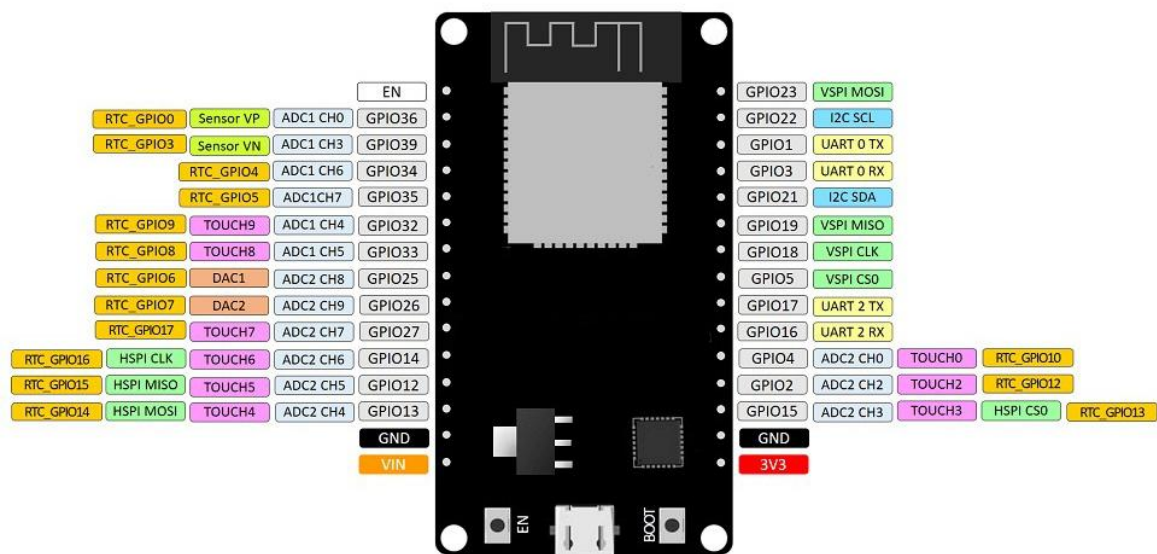


۳-۲- توضیحات مدار چاپی

در شکل زیر، ترمینال‌های موجود در مدار چاپی با شماره معرفی شده‌اند:

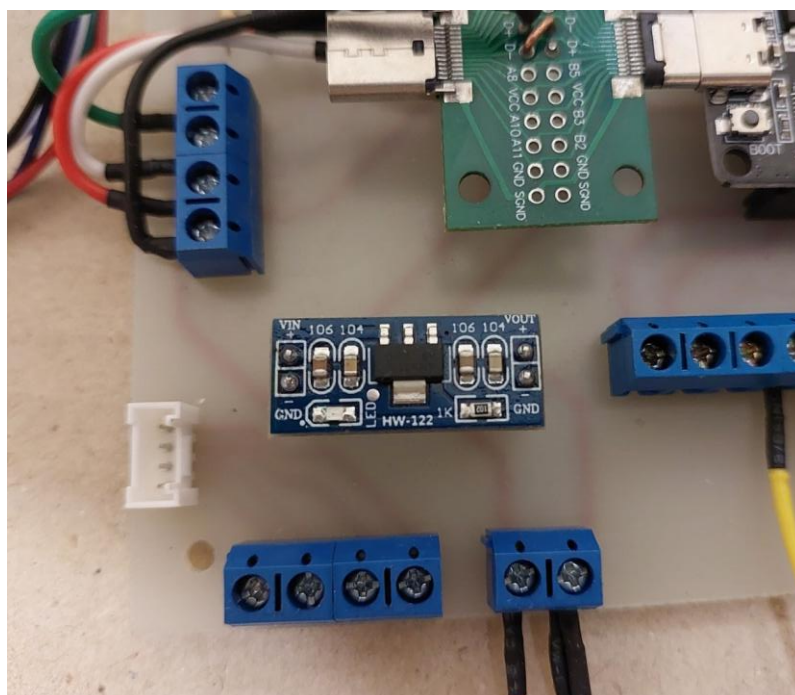


در مدار ساخته شده، برد ESP32 وارد پین‌های هدر شده و هر پین آن به یک ترمینال متصل شده است. شماره‌های ۱ الی ۳۰ دقیقاً متناظر با پین‌های برد ESP32 است. پین‌های ESP32 به طور دقیق به صورت شکل زیر است:



همچنین در این مدار، یک رگولاتور AMS117 3.3v وجود دارد برای تامین ولتاژ ۳.۳ ولت کمکی. در این مدار تمام زمین‌ها به هم متصل است و یک زمین مشترک برای تمام قطعات ایجاد شده است. از رگولاتور تنها برای تامین جریان لازم LCD استفاده میشود تا فشار کمتری به رگولاتور خود برد ESP32 تحمیل شود. خروجی رگولاتور به دو ترمینال متصل است که ترمینال ۴۸ ولتاژ ۳.۳ ولت و ترمینال ۴۷ گراند است.

توجه: رگولاتور باید به گونه ای وارد هدرها شود که ترمینال ۴۷ زمین شود. به عبارتی اگر از بالا به برد نگاه کنیم ، زمین GND مازول رگولاتور باید پایین باشد، تمام مسیره‌های زمین نیز به هم متصل است. به صورت شکل زیر:



ولتاژ این دستگاه، از طریق کابل USB Type-C تامین میشود. این دستگاه را میتوان به هر نوع شارژر موبایل وصل کرد، چه کابل دو سر Type-C باشد، چه یک سر Type-C و سر دیگر Type-A باشد. یا حتی میتوان آن را به کامپیوتر یا لپتاپ وصل کرد. در صورت اتصال به کامپیوتر یا لپ تاپ، علاوه بر تامین ولتاژ روشن شدن دستگاه، امکان برنامه نویسی یا ذخیره دیتا نیز فراهم میشود. برای اینکه بتوان از USB Type-C استفاده کرد، لازم است روی دستگاه سوکت مادگی Type-C قرار داده شود. کابل های USB را میتوان به دو دسته کلی تقسیم کرد، کابل هایی که دو سر Type-C هستند، و کابل هایی که یک سر تایپ A و سر دیگر تایپ C است. ساختار و پروتکل های این دو دسته کاملاً با هم فرق دارد.

اگر یک سر کابل تایپ A باشد (چه نسخه ۲ چه نسخه ۳)، اگر سمت تایپ سی آن به دستگاه ما یا ESP32 متصل شود و سر دیگر آن که تایپ A است به هر نوع شارژر یا کامپیوتر و .. متصل شود، به طور قطعی تنها ولتاژ ۵ ولت به آن متصل شده و بدون هیچ مشکلی دستگاه کار میکند. بنابراین میتوان از این نوع کابل برای کار با دستگاه استفاده کرد.

اما در دسته دوم که دو سر تایپ سی است، داستان متفاوت است. از این کابل های مدرن برای فست شارژ و انتقال دیتای سنگین و .. استفاده میشود. در این کابل ها، در حالت استاندارد دو سیم اضافه به نام Configuration Channels وجود دارد تا دستگاه مبدا و مقصد از طریق آن با هم صحبت میکنند. اگر

از کابل دو سر تایپ سی استفاده شود برای دستگاه یا ESP32، باید به از طریق سیم های CC به آن بفهمانیم که ولتاژ ۵ ولت میخواهیم، تا شارژر آن را تامین کند. برای این کار، لازم است سیم های CC از طریق مقاومت های ۵.۱ کیلو اهمی، به گراند متصل شوند. در این حالت، ولتاژ ۵ ولت اعمال میشود. اگر این کار انجام نشود، کلا شارژر هیچ ولتاژی اعمال نمیکند. به همین دلیل روی مدار چاپی، این دو مقاومت قرار داده شده تا دستگاه با هر نوع کابلی و با هر نوع شارژر و دستگاهی سازگار شود. اطلاعات فنی در فولدر Resources قرار دارد. (فایل Overview of USB Type-C....PDF)

قابل ذکر است در برخی سوکت های پنبلی تایپ سی، دو سیم CC موازی شده است که لازم است همان یک سیم از طریق مقاومت به گراند متصل شود. سوکتی که در این پروژه استفاده شده به همین صورت است. اما برای تنظیمات پیشرفته تر ۲ سیم CC نیاز است که در مدار چاپی تعبیه شده است.

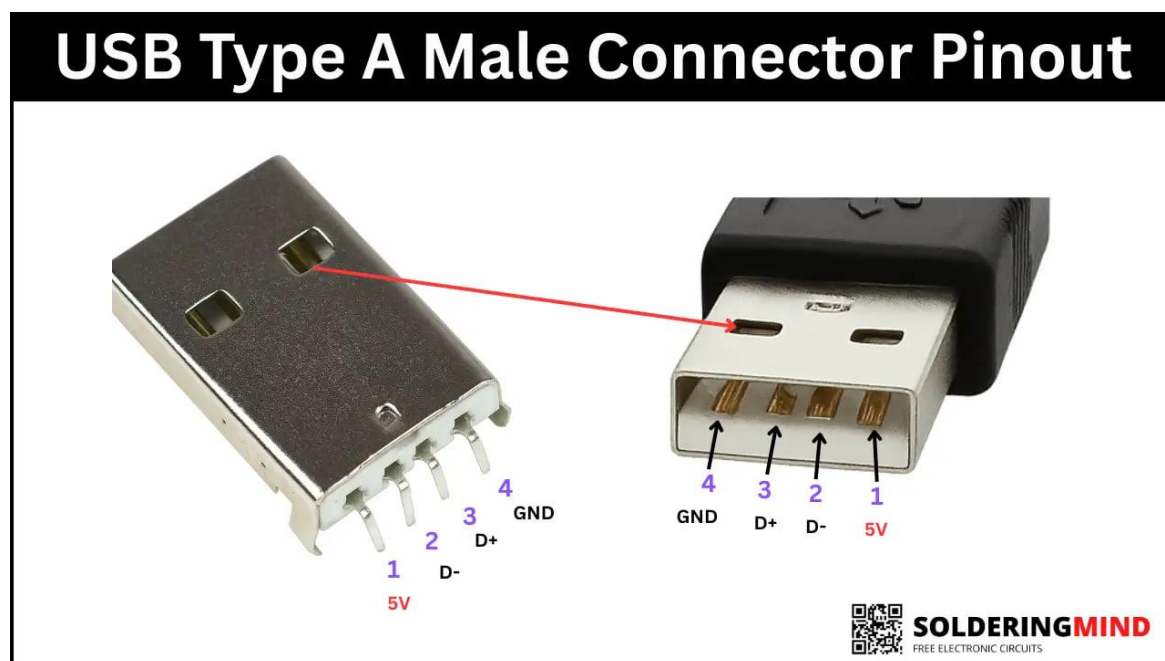
همچنین یک دیود شاتکی 1N5817 در مسیر ولتاژ ۵ ولت قرار دارد تا برای حفاظت از دستگاه اصلی (کامپیوتر و لپتاپ و ...)، مانع از برگشت جریان شود، بخصوص مواقعی که از منبع تغذیه خارجی استفاده میشود. برای این کار، میتوان از دیود های 1N5818 و 1N5819 نیز استفاده کرد، اما 1N5817 افت ولتاژ کمتری ایجاد میکند و اولویت با آن است.

ترمینال های ۳۱ الی ۳۶ مربوط به سیم های سوکت تایپ سی پنبلی است. گراند سوکت به ترمینال ۳۱ متصل میشود. VCC سوکت که همان ولتاژ ۵ ولت است، به ترمینال ۳۶ متصل میشود. سیم CC به ترمینال ۳۲ یا ۳۳ متصل میشود. (اگر دو سیم CC بود هر کدام را به یکی متصل میکنیم، فرقی ندارند). سیم دیتای مثبت D+ سوکت تایپ سی را به ترمینال ۳۴ و سیم دیتای منفی D- را به ترمینال ۳۵ متصل میکنیم.

یک کلید راگر برای خاموش و روشن کردن دستگاه (یا به عبارتی ریست) وجود دارد که یک سیم آن را به ترمینال ۳۷ و سیم دیگر آن را به ترمینال ۳۸ میندیم. فرقی در جهت دارد، صرفاً قطع و وصل میکند. برای ارتباط سوکت تایپ سی پنبلی با برد ESP32، لازم است سیم های تغذیه و دیتا به برد برسد، برای اینکار از برد Break out تایپ سی استفاده شده است. در این برد برک اوت، تمام پین ها روی PCB آن وجود دارد و میتوان به آن سیم لحیم کرد. باید توجه کرد که با توجه به اینکه سوکت تایپ سی متقارن است، هر پین دوتاست، برای اینکه از هر دو جهت کار کند، لازم است پین های متناظر اتصال کوتاه شوند. بنابراین پین های D+, VCC, GND و D- اتصال کوتاه شده اند. سپس از هر کدام سیمی بیرون

آمده تا از طریق ترمینال های ۳۹ الی ۴۲ به سیم های سوکت تایپ سی پنبلی متصل شوند. سیم دیتای مثبت برک اوت D+ به ترمینال ۳۹ متصل شده و دیتای منفی D- برک بوت به ترمینال شماره ۴۰ متصل میشود. همچنین VCC به ترمینال ۴۱ و گراند به ترمینال ۴۲ متصل میشوند.

نکته مهم: در بسیاری از سوکت ها و سیم های USB، رنگ سیم ها استاندارد نیست. بنابراین برای شناسایی دیتاها و تغذیه و ... نباید به رنگ سیم ها توجهی داشت، باید شناسایی شوند. برای شناسایی میتوان از کابل یک سر تایپ A استفاده کرد. وقتی کابل از روبرو دیده شود، به گونه ای که پلاستیک آن به طرف پایین باشد، پین های آن به صورت تصویر زیر است (این مورد قطعا استاندارد خواهد بود)



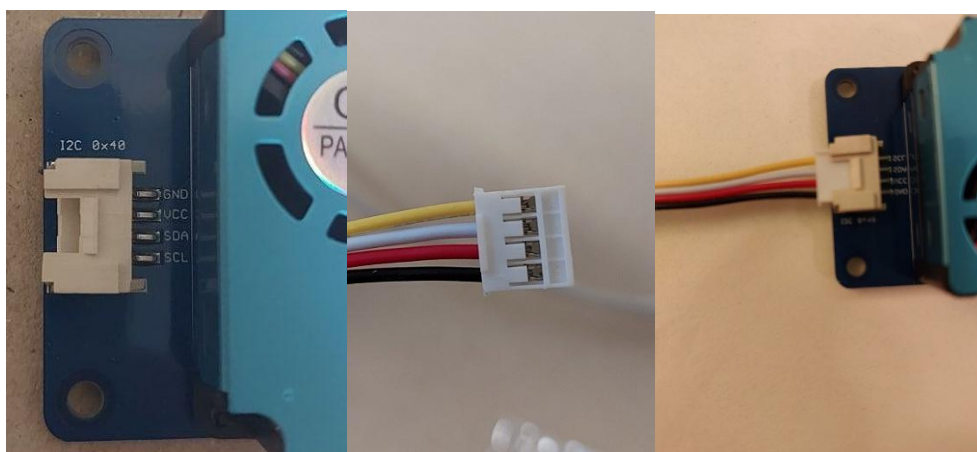
سپس سر دیگر این کابل را به سوکت تایپ سی میزنیم و با اهم متر، اتصال کوتاه بین سیم ها و پین های تایپ A را بررسی میکنیم. همانطور که در این پروژه میبینیم، در سوکت استفاده شده، دیتای مثبت رنگ ابی دارد و CC رنگ سبز، که غیر استاندارد است!

ترمینال ۴۳ الی ۴۶ برای سنسور است. صرفا به صورت کمکی این ترمینال ها قرار داده شده اند تا مستقیما کانکتور سنسور روی برد قرار گیرد. اما چون سیم کوتاه بود، در اینجا من سیم ها رو بلند کردم و مستقیما

به ترمینال های دیگر متصل کردم. با این حال میتوان مستقیماً سیم سنسور را به کانکتور تعبیه شده متصل کرد، سپس از ترمینال های ۴۳ الی ۴۶ سیم کشید به سایر ترمینال ها. در این حالت، ترمینال ۴۶ گراند خواهد بود، ترمینال ۴۵ ولتاژ ۵ ولت، ترمینال ۴۴ پین SDA و ترمینال ۴۳ پین SCL خواهد بود. همواره اتصال سیم ها و منطق آنها چندبار چک شود، تا اشتباهات خسارتی ایجاد نشود.

۳-۳- اتصال پایه ها

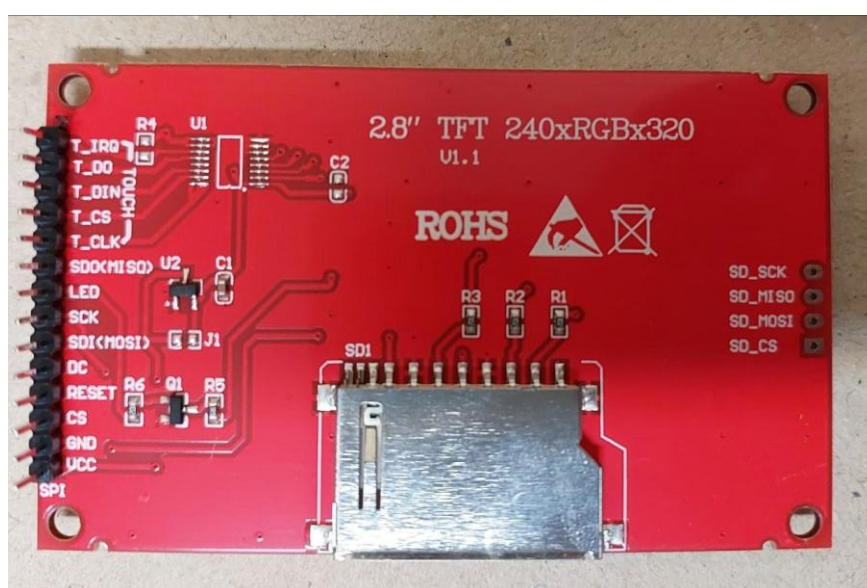
سنسور HM3301 مبتنی بر پروتکل I2C است که ۴ پین (سیم) دارد. این ۴ پین، مطابق جدول زیر به برد وصل میشوند:



عنوان پین	رنگ سیم	توضیحات	اتصال به
GND	مشکی	زمین مشترک	ESP32: GND
VCC	قرمز	ولتاژ تغذیه	ESP32: VIN
SDA	سفید	Serial Data	ESP32: GPIO21
SCL	زرد	Serial Clock	ESP32: GPIO22

نکته: برای سادگی کار، میتوان مستقیماً سوکت I2C سنسور را به هدر تعبیه شده روی مدار متصل کرد، سپس از ۴ ترمینال متناظر، سیم کشید و به ترمینال های ESP32 متصل کرد. اما به دلیل کوتاه بودن طول این سیم، افزایش طول انجام شده است و سیم‌ها مستقیماً به ترمینال‌های ESP32 متصل شده‌اند.

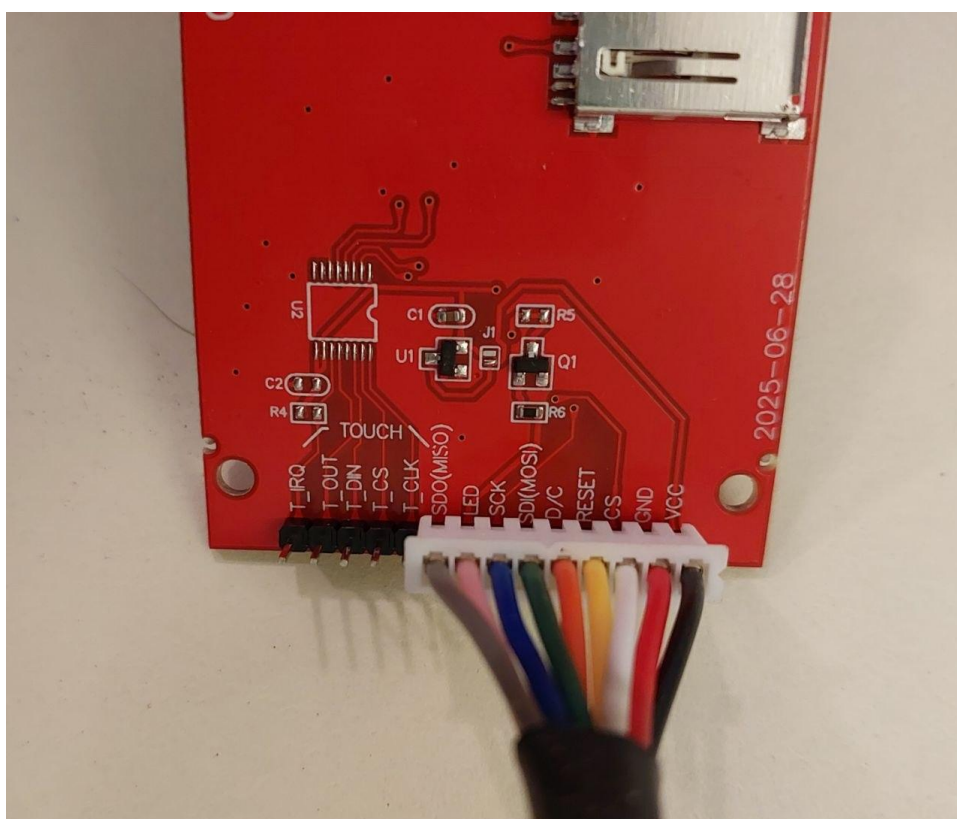
ال سی دی استفاده شده از نوع TFT با درایور ST7789 و رابط SPI است. پین‌های این نوع LCD به صورت زیر است:



به صورت پیشفرض پین هدر در روی LCD نصب شده است. برای اتصال این پین‌ها به ترمینال‌ها، از کانکتور ۹ پین استفاده شده است که به صورت زیر است:



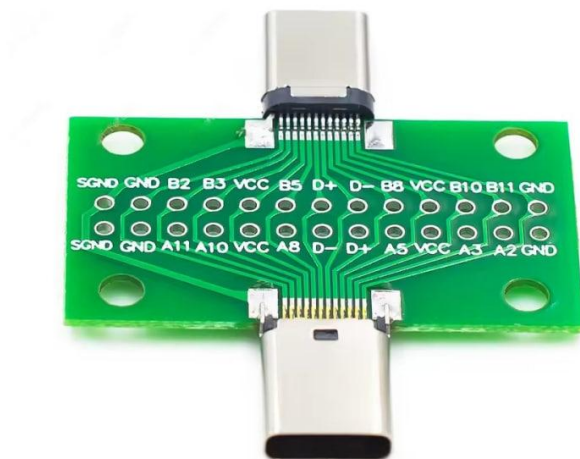
در LCD پنچ پین وجود دارد که مربوط به تاچ است و با حرف T شروع شده‌اند، اما این LCDها تاچ ندارند، صرفاً برای تولید، از یک برد مادر برای انواع مختلف LCD استفاده میشود و در اینجا این پین‌ها به جایی وصل نیستند. بنابراین حداکثر ۹ پین از ۱۴ پین موجود کارایی دارند. رنگ سیم‌ها کانکتور به صورت استاندارد با پین‌ها LCD سازگاری ندارند. اما در این پروژه، کانکتور به صورت قراردادی به صورت به LCD متصل میشود که سیم مشکی، به VCC متصل شود. دقت شود که رنگ قرمز و مشکی، در اینجا برعکس شده است. پین‌های LCD به صورت جدول زیر، به ترمینال‌ها متصل میشود.



عنوان پین	رنگ سیم	توضیحات	اتصال به
SDO (MISO)	توسی	خروجی داده نمایشگر	ESP32: GPIO19
LED	صورتی	نور پس زمینه	Regulator: VOUT (T: 48)
SCK	آبی	کلاک سیگنال SPI	ESP32: GPIO18
SDI (MOSI)	سبز	ورودی داده نمایشگر	ESP32: GPIO23
D/C	نارنجی	Data/Command	ESP32: GPIO4
RESET	زرد	ریست سخت افزاری LCD	ESP32: GPIO2
CS	سفید	Chip Select	ESP32: GPIO5
GND	قرمز	زمین مشترک	Regulator: GND (T: 47)
VCC	مشکی	ولتاژ تغذیه	Regulator: VOUT (T: 48)

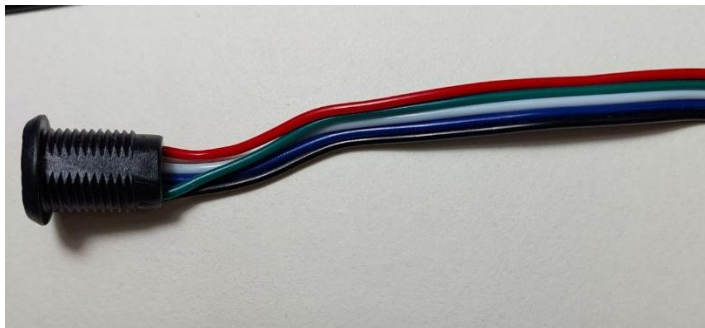
توجه شود کانکتور به درستی به پین های هدر ال سی دی متصل شود. سیم مشکی به VCC میرسد که کاملاً غیر استاندارد است.

پایه‌های برد برک اوت Type-C که مستقیماً به ESP32 متصل شده است طبق استاندارد به صورت جدول زیر است:



عنوان پین	رنگ سیم	توضیحات	اتصال به
GND	مشکی	زمین مشترک	T: 42
VCC	قرمز	ولتاژ تغذیه (۵ ولت)	T: 41
D+	سبز	دیتا مثبت	T: 39
D-	سفید	دیتا منفی	T: 40

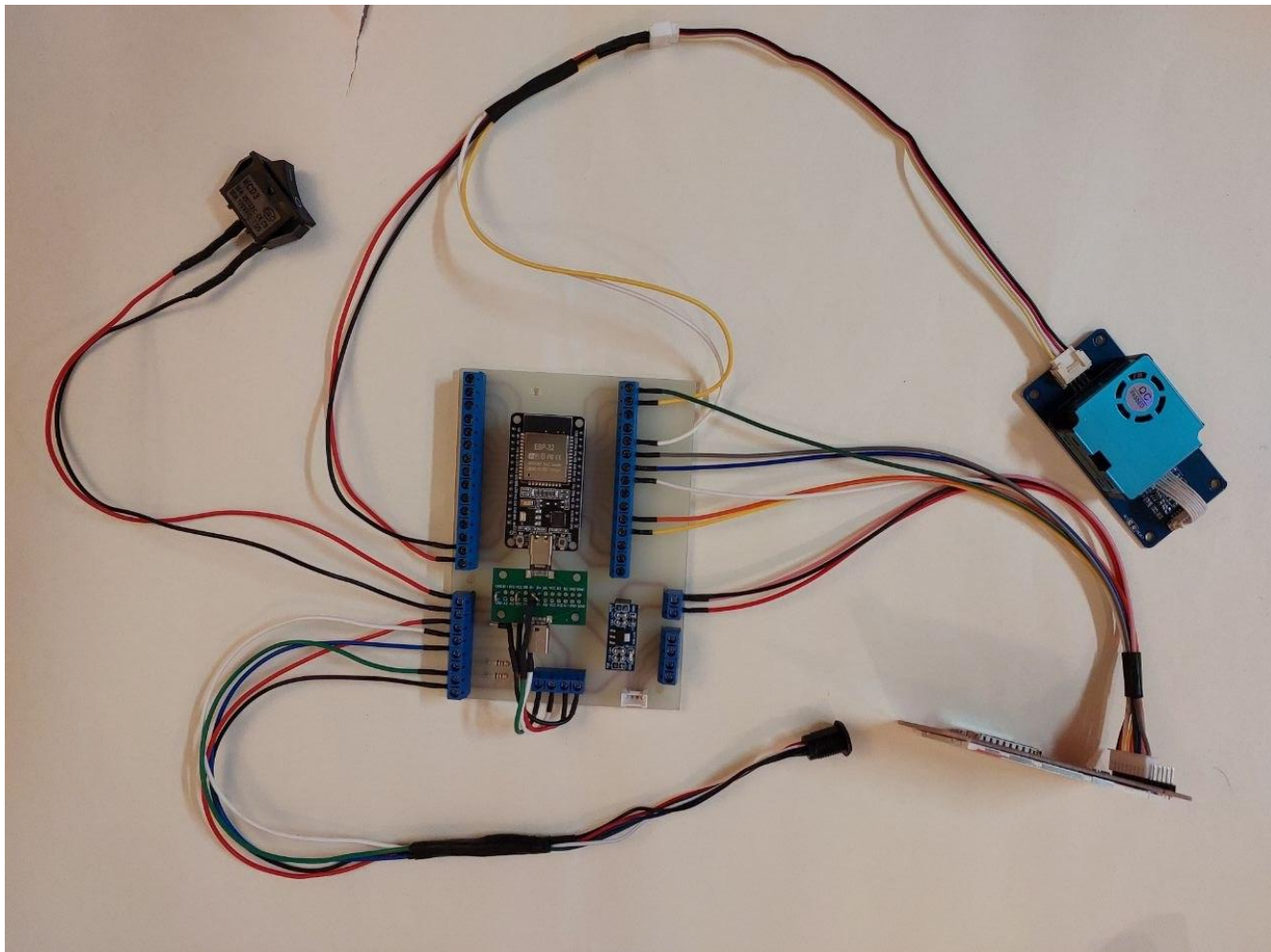
سیم‌های کانکتور Type-C پنبلی نصب شده روی باکس، شامل ۵ سیم به شرح زیر است:

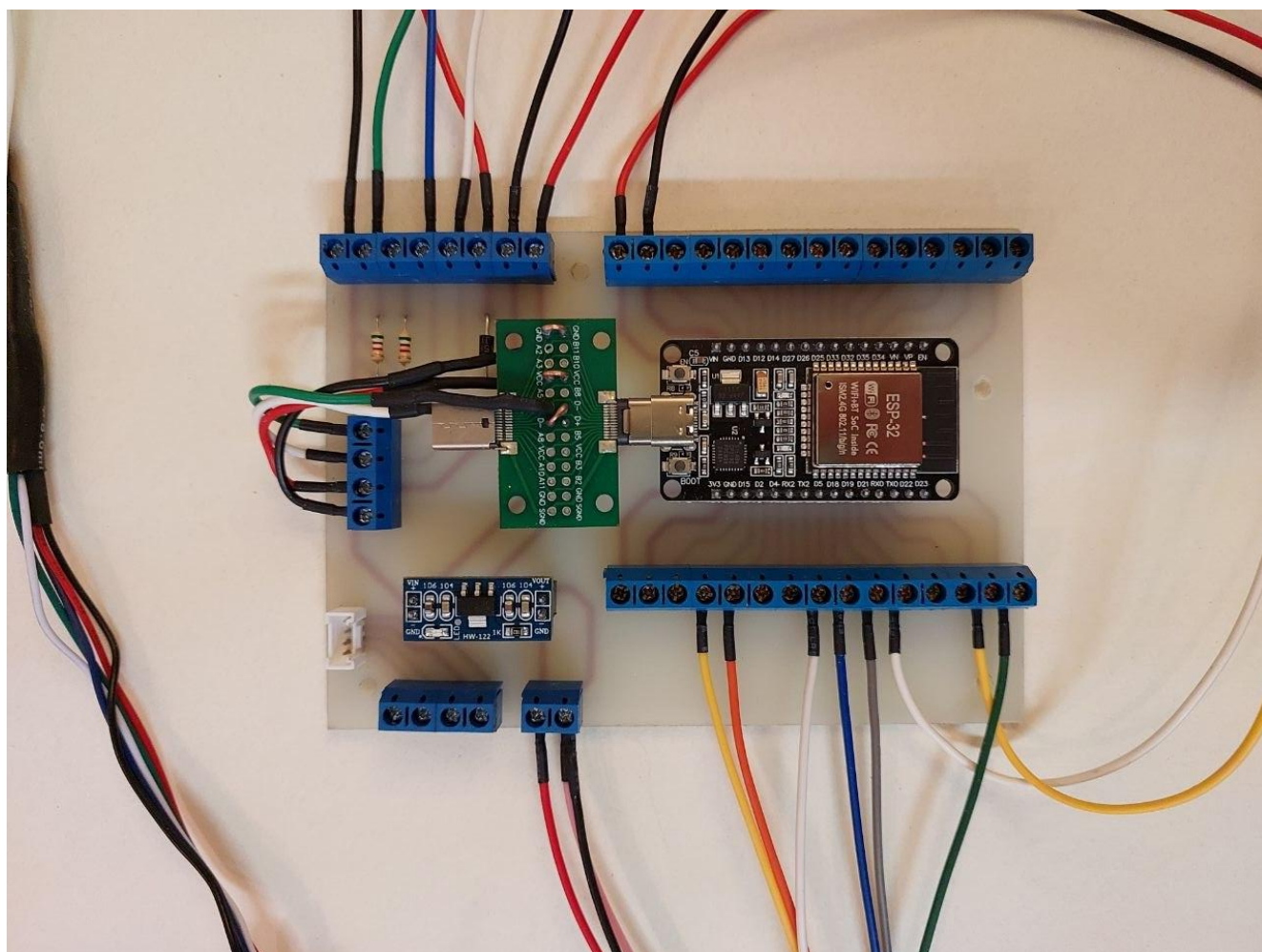


عنوان پین	رنگ سیم	توضیحات	اتصال به
GND	مشکی	زمین مشترک	T: 31
VCC	قرمز	ولتاژ تغذیه (۵ ولت)	T: 36
D+	آبی	دیتا مثبت	T: 34
D-	سفید	دیتا منفی	T: 35
CC	سبز	Configuration Channel	T: 32

نکته: در حالت ایده‌آل، دو سیم CC وجود دارد که روی مدار چاپی، جای آن تعبیه شده است. با این حال در این پروژه از مدل ۵ سیم استفاده شده است که دو پین CC در داخل سوکت Type-C موازی شده‌اند. همین مدل ۵ سیم، برای این پروژه کاملاً کافی است. اما اگر قصد تغییر این سوکت را داشته باشید، همه چیز به حالت استاندارد طراحی شده است و امکان هر کاری وجود دارد.

تصویر مدار بسته شده به صورت زیر است (توجه، رنگ سیم های ال سی دی استاندارد نیست)





همانطور که در تصویر بالا مشاهده میشود، گراند های مازول رگولاتور پایین هستند.

۳-۴- توضیحات تکمیلی

در بردهای ESP32 سیگنال‌های فرمان ۳.۳ ولت هستند. برخی از بردهای آردوینو نیز ۳.۳ هستند، اما بسیاری از بردهای معروف آردوینو مثل مدل Uno از سطح ولتاژ ۵ ولت برای سیگنال فرمان استفاده میکنند. در این پروژه، ال سی دی نیاز به ولتاژ تغذیه و فرمان ۳.۳ ولت دارد. بنابراین انتخاب برد ۳.۳ ولت بسیار مناسب تر است. اگر برد ۵ ولت استفاده شود باید با استفاده از قطعات و تکنیک‌های خاص، سطح ولتاژ فرمان به ۳.۳ ولت کاهش پیدا کند.

در بردهای ESP32 و آردوینو، تغذیه برد به چند روش میتواند انجام شود. یعنی برد روشن شود و کار کند. در این بردها، یک سوکت برای اتصال سریال به کامپیوتر وجود دارد که اغلب USB هستند. مثلاً در مورد برد ESP32 ما، پورت USB Type-C وجود دارد. در برخی مدل‌های آردوینو، علاوه بر این پورت، یک Barrel Jack وجود دارد که از آداپتورهای ۵ ولت یا ۳.۳ ولت (متناسب با نوع برد) میتوان برای تامین ولتاژ تغذیه استفاده کرد.

علاوه بر این روش‌ها، در حالت تخصصی، به چند روش دیگر هم میتوان تغذیه بردها را تامین کرد. در این برد‌ها، مثل همین ESP32 و آردوینو، علاوه بر پورت تایپ سی و ...، پین‌های VIN و 3V3 وجود دارد. در برد ESP32 پین VIN مستقیماً متصل به VCC پورت تایپ سی است. یعنی اگر از کابل USB استفاده نشود، میتوان ولتاژ ۵ ولت را به پایه VIN برد متصل شد و گراند را به GND و برد مثل حالت قبل روشن میشود. قابل ذکر است تمام گراند‌ها در تمام بردها به هم متصل است. صرفاً دلیل چندتایی بودن آن، استفاده ساده تر است.

حال این ولتاژ ۵ ولت، از طریق USB یا پین VIN تامین شود، با استفاده از رگولاتور AMS1117 3V3 که روی برد ESP32 است، تبدیل به ۳.۳ ولت میشود و مدار روشن شده و کار میکند. روش دیگر روشن کردن مدار این است که مستقیماً به پین 3V3 ولتاژ ۳.۳ ولت متصل کرد و گراند را به GND. در این حالت نیز مدار روشن شده و عالی کار میکند و دیگر جریان از رگولاتور نمی‌گذرد. بنابراین به روش‌های مختلف میتوان برد را روشن کرد. در آردوینو نیز به همین شکل است. میتوان با اعمال ولتاژ به پین‌ها، آن را روشن کرد. این روش فقط برای روشن کردن برد است و دیگر امکان برنامه نویسی روی آن وجود ندارد، زیرا پینی برای انتقال دیتای سریال وجود ندارد. برای انتقال دیتای سریال، نیاز به مبدل‌های سریالی

هست و فقط از طریق پورت USB میتوان دیتا را عبور داد. در برد ESP32 مبدل سریال ما CP2102 بود.

نکته ای که وجود دارد این است که فقط از یک طریق باید ولتاژ تامین شود. یعنی اگر با کابل USB برد روشن شده، هرگز نباید ولتاژ ۵ ولت را به پایه ان زد یا ۳.۳ ولت را به پایه 3V3. در این حالت مشکلات جدی ایجاد شده و برد و دستگاه اصلی و .. آسیب جدی میبیند. به همین دلیل یک دیود در مسیر ولتاژ قرار داده شده است. با این حال برای این دستگاه، میتوان به سادگی ولتاژ را به پایه ها متصل کرد و دستگاه کار کرده و اپلیکیشن ها نیز بدون هیچ تغییری کار خواهند کرد.

با توجه به این توضیحات، قابل ذکر است وقتی کابل USB به برد وصل است، پایه VIN ولتاژ ۵ ولت خواهد داشت و از طریق رگولاتور روی برد، پایه 3V3 ولتاژ ۳.۳ ولت.

نکته دیگری که وجود دارد، این است که در این پروژه، به دو دلیل تغذیه سنسور HM3301 به ولتاژ ۵ ولت VIN متصل است. دلیل اول و مهم این است که اگر از ولتاژ ۳.۳ ولت برد استفاده میشد، جریان مورد نیاز سنسور از رگولاتور روی برد رد میشد و این فشار زیادی به رگولاتور برد می آورد. این رگولاتور باید بار سبکی داشته باشد تا بتواند جریان بلوتوث را تامین کند. بنابراین از ۵ ولت که در پایه VIN است برای سنسور استفاده شده است. همچنین در منابع بیشتر توصیه شده که از ۵ ولت برای این سنسور استفاده شود تا فن و عملکرد سنسور بهتر باشد.

در مورد سنسور، ولتاژ تغذیه به ۵ ولت وصل است، اما سیگنال های فرمان آن که به ESP32 متصل است، ۳.۳ ولت هستند. این مساله هیچ مشکلی ایجاد نمیکند و این به دلیل ساختار پروتکل I2C است که میتواند سطح ولتاژ تغذیه و سیگنال های فرمان متفاوت باشد. در دیتاشیت سنسور نیز همین سطوح ولتاژ پیشنهاد شده است.

دقیقا به همین دلیل که به رگولاتور برد ESP32 فشار زیادی وارد نشود، برای ال سی دی که جریان بالایی مصرف میکند نیز از رگولاتور مستقلی استفاده شده است، که گراند آن به گراند برد متصل است. در این حالت سطح ولتاژ ۳.۳ ولت سیگنال های فرمان SPI با ۳.۳ ولتی که از رگولاتور مستقل تامین میشود، هماهنگ خواهند بود و بدون مشکل کار خواهند کرد.

نکته بعدی در این سیستم، آن است که تمام گراند‌ها به هم متصل است. در اکثر سیستم‌ها، حتی اگر چند منبع تغذیه وجود داشته باشد، امکان مشترک کردن گراند‌ها برای گرفتن ولتاژهای هم سطح امکان پذیر است.

در مورد PCB اختصاصی ساخته شده، گراند‌ها به هم متصل است. مسیرهای VCC و GND ضخیم تر هستند تا هم جریان به خوب عبور کند و هم افتا ولتاژ کاهش یابد (به دلیل کاهش مقاومت مسیر).

همچنین پشت این برد از Solder Mask استفاده نشده. سولدر مسک همان رنگ محافظ سبز، ابی یا قرمز است که روی بردها دیده میشود. در عوض برای حفاظت از مسیرها و بهبود کیفیت، از اسپری پلاستیک ۷۰ استفاده شده است که برای همین کار ساخته شده است و همان نقش سولدر مسک را ایفا میکند.