

دانشکده فنی

پروژه کارشناسی

نیمسال دوم سال تحصیلی 97-98

**رشته و گرایش**

مهندسی کامپیوتر - نرم افزار

**عنوان پروژه**

کنترل سوئیچ از راه دور

**دانشجویان**

سید حمید حسینی نژاد (9312268110)

صدرا صمدی (9312268122)

**استاد راهنما**

دکتر سید محمد حسین شکریان

فهرست

[مقدمه 4](#_Toc32770836)

[اینترنت اشیاء 4](#_Toc32770837)

[تاریخچه اینترنت اشیاء (IOT) 4](#_Toc32770838)

[کاربرد اینترنت اشیاء 5](#_Toc32770839)

[کاربرد اینترنت اشیاء در خانه هوشمند 5](#_Toc32770840)

[کاربرد اینترنت اشیاء در نگهداری از سالمندان 5](#_Toc32770841)

[کاربرد اینترنت اشیاء در خدمات پزشکی و بهداشتی 6](#_Toc32770842)

[کاربرد اینترنت اشیاء در اتوماسیون ساختمان و خانه 6](#_Toc32770843)

[کاربردهای صنعتی اینترنت اشیاء 6](#_Toc32770844)

[کاربرد اینترنت اشیاء در کشاورزی 6](#_Toc32770845)

[برخی از تکنولوژی های اینترنت اشیا 7](#_Toc32770846)

[آردوینو 7](#_Toc32770847)

[ماژول‌ها و شیلدهای آردوینو 9](#_Toc32770848)

[آردوینو nano 10](#_Toc32770849)

[مشخصات برد آردوینو nano 11](#_Toc32770850)

[ماژول WiFi 12](#_Toc32770851)

[کاربردهای ماژول ESP8266 12](#_Toc32770852)

[انواع ماژول ESP8266 13](#_Toc32770853)

[حالت های ماژول ESP8266 13](#_Toc32770854)

[در نقش Access Point 14](#_Toc32770855)

[در نقش Station 14](#_Toc32770856)

[حالت ترکیبی Station و Access Point 15](#_Toc32770857)

[Platform IO 15](#_Toc32770858)

[برنامه نویسی سرور 16](#_Toc32770859)

[معرفی نود جی اس (Node.js) 16](#_Toc32770860)

[کاربردهای نود جی اس (Node.js) 17](#_Toc32770861)

[ویژگی های نود جی اس (Node.js) 17](#_Toc32770862)

[RxJS 18](#_Toc32770863)

[Express 18](#_Toc32770864)

[socket.io 18](#_Toc32770865)

[نحوه کار برنامه سرور 19](#_Toc32770866)

[برنامه نویسی پنل کنترل کاربری اندروید 21](#_Toc32770867)

[کاتلین (Kotlin) 21](#_Toc32770868)

[ویژگی های کاتلین 21](#_Toc32770869)

[نحوه کار کد اندروید 22](#_Toc32770870)

[تصویر نهایی برنامه 24](#_Toc32770871)

[نصب برنامه 24](#_Toc32770872)

[برنامه نویسی برد آردوینو 24](#_Toc32770873)

[ArduinoJson 24](#_Toc32770874)

[نحوه کار برنامه برد آردوینو 25](#_Toc32770875)

[نحوه کار کد برد آردوینو 25](#_Toc32770876)

[برنامه نویسی ماژول WiFi 28](#_Toc32770877)

[ArduinoJson 28](#_Toc32770878)

[Ticker 28](#_Toc32770879)

[WebSockets 29](#_Toc32770880)

[SocketIOClient 29](#_Toc32770881)

[WiFiManager 29](#_Toc32770882)

[نحوه کار کد ماژول WiFi 30](#_Toc32770883)

[شمای کلی پروژه 34](#_Toc32770884)

[شمای کلی برد سخت افزاری 35](#_Toc32770885)

[تصویر نهایی برد سخت فزاری 36](#_Toc32770886)

[منابع 37](#_Toc32770887)

# مقدمه

گرچه شروع اینترنت اشیاء را می‌توان به سی سال پیش نسبت داد اما فراگیری بی سابقه این فناوری در سال‌های اخیر و پیوند آن با فناوری‌های نوین باعث شده تصور آینده بدون این تکنولوژی ناممکن شود. از طرفی با توجه به اهمیت نحوه نگهداری و گردش اطلاعات در جهان، لازم است تلاش‌های جدی برای بومی سازی این فناوری صورت گیرد تا امنیت مصرف کنندگان به خطر نیفتد و در صورت اعمال تحریم ارائه خدمات به مصرف کنندگان قطع نشود.

همچنین همزمانی فراگیری اینترنت اشیا با ظهور روش‌های نوین پردازش داده و هوش مصنوعی و ارائه رایانش ابری در بسترهای داخلی و خارجی باعث شده با ترکیب فناوری‌های یاد شده، فرصت‌های نوین نوآوری در حفاظت محیط زیست، سلامت، اتوماسیون و کشاورزی و دام‌داری ایجاد کند.

در این پروژه سعی شده یک مدل اولیه اینترنت اشیا یک کلید با استفاده از پنل کاربری اندروید کنترل شود.

# اینترنت اشیاء

Internet of Things یا به اختصار IoT سیستمی از دستگاه های محاسباتی بهم پیوسته، ماشین های مکانیکی و دیجیتال، اشیاء، حیوانات یا افراد است که با شناسه های منحصر به فرد (UID) و امکان انتقال داده از طریق شبکه بدون نیاز به تعامل انسانی یا تعامل انسان با کامپیوتر، ارائه می شود.

تعریف اینترنت اشیاء به دلیل همگرایی چندین فناوری، تجزیه و تحلیل در زمان واقعی، یادگیری ماشین، حسگر اشیا و سیستم های تعبیه شده تکامل یافته است. زمینه های سنتی سیستم های تعبیه شده، شبکه های حسگر بی سیم، سیستم های کنترل، اتوماسیون (از جمله اتوماسیون در منزل و ساختمان) و سایر موارد در ایجاد اینترنت اشیاء نقش دارند. در بازار مصرف، فناوری IoT مترادف ترین محصولاتی است که به مفهوم "خانه هوشمند" مربوط می شود و وسایل را پوشش می دهد (مانند وسایل روشنایی، ترموستات، سیستم های امنیتی خانگی و دوربین ها و سایر لوازم خانگی) که از اکوسیستم های متداول تر و از طریق دستگاه های مرتبط با آن اکوسیستم مانند تلفن های هوشمند و بلندگوهای هوشمند قابل کنترل است.

اینترنت اشیاء را می توان به عنوان شبکه ای از اشیاء فیزیکی یا افراد به نام "چیزهایی" تعریف کرد که با نرم افزار، الکترونیک، شبکه و حسگرها تعبیه شده اند که به این اشیاء امکان جمع آوری و تبادل داده ها را می دهد. هدف IoT گسترش اتصال به اینترنت از دستگاه های استاندارد مانند رایانه، موبایل، تبلت به دستگاه های نسبتاً گنگ مانند توستر است. IoT با بهبود جنبه های زندگی ما با قدرت جمع آوری داده ها، الگوریتم های هوش مصنوعی و شبکه ها، تقریبا همه چیز را "هوشمند" می کند. ویژگی های موجود در IoT همچنین می تواند برای شخص دارای کاشت مانیتور دیابت، حیوان با دستگاه ردیابی و غیره استفاده شود.

# تاریخچه اینترنت اشیاء (IOT)

مفهوم شبکه دستگاه های هوشمند از اوایل سال 1982 مورد بحث قرار گرفت، که دستگاه فروش خودکار در دانشگاه کارنگی ملون تبدیل به اولین دستگاه متصل به اینترنت شد که توانست گزارش موجودی خود را ارائه دهد و اینکه آیا نوشیدنی های تازه بارگیری شده سرد بوده یا خیر. مقاله مارک ویزر در سال 1991 در مورد رایانه همه جا، "کامپیوتر قرن 21"، و همچنین مکان های دانشگاهی مانند UbiComp و PerCom چشم انداز معاصر IoT را تولید کرد. در سال 1994، رضا راجی مفهوم موجود در IEEE Spectre را به عنوان "انتقال بسته های کوچک داده به مجموعه بزرگی از گره ها، به منظور ادغام و اتوماسیون همه چیز از لوازم خانگی به کل کارخانه ها" توصیف کرد. بین سال های 1993 و 1997، چندین شرکت راه حل هایی مانند مایکروسافت را در محل کار یا شرکت نول ارائه دادند. هنگامی که بیل جوی ارتباط بین دستگاه به دستگاه را به عنوان بخشی از چارچوب "شش شبکه" خود، که در سال 1999 در مجمع جهانی اقتصاد در داووس ارائه شد، تصور کرد.

اصطلاح "اینترنت چیزها" احتمالاً در سال 1999 توسط کوین اشتون از Procter & Gamble، بعداً مرکز خودکار شناسایی MIT ابداع شده است، اگرچه این عبارت را "اینترنت برای چیزها" ترجیح می دهد. در آن مرحله، وی شناسایی فرکانس رادیویی (RFID) را برای اینترنت اشیا ضروری دانست، که به کامپیوترها امکان می دهد همه چیزهای شخصی را مدیریت کنند.

# کاربرد اینترنت اشیاء

مجموعه گسترده برنامه های کاربردی برای دستگاه های IoT اغلب به فضاهای مصرفی، تجاری، صنعتی و زیربنایی تقسیم می شود.

بخش فزاینده ای از دستگاه های IoT برای کاربرد مصرف کننده ایجاد می شود، از جمله وسایل نقلیه متصل، اتوماسیون منزل، فناوری پوشیدنی، بهداشت متصل و وسایل دارای قابلیت نظارت از راه دور.

# کاربرد اینترنت اشیاء در خانه هوشمند

دستگاه های IoT بخشی از مفهوم بزرگتر اتوماسیون منزل است که می تواند شامل روشنایی، گرمایش و تهویه مطبوع، رسانه ها و سیستم های امنیتی باشد. مزایای بلند مدت می تواند با صرفه جویی خودکار چراغ ها و الکترونیک، موجب صرفه جویی در مصرف انرژی شود.

خانه هوشمند یا خانه اتوماتیک می تواند مبتنی بر سکوی یا مراکز مستقر در کنترل دستگاه ها و لوازم هوشمند باشد. به عنوان مثال، با استفاده از HomeKit اپل، تولید کنندگان می توانند محصولات خانگی و لوازم جانبی خود را توسط برنامه در دستگاه های iOS مانند iPhone و Apple Watch کنترل کنند. این می تواند برنامه اختصاصی یا برنامه های بومی iOS مانند Siri باشد. این امر را می توان در مورد Smart Home Essentials Lenovo نشان داد، که این مجموعه ای از دستگاه های خانگی هوشمند است که بدون نیاز به پل Wi-Fi از طریق اپلیکیشن Home Apple یا Siri کنترل می شوند. همچنین مراکز خانگی هوشمند اختصاصی وجود دارد که به عنوان سیستم عامل های مستقل برای اتصال محصولات مختلف خانه هوشمند ارائه می شوند و از جمله آنها می توان به Amazon Echo ،Google Home ، HomePod اپل و SmartThings Hub سامسونگ اشاره کرد. علاوه بر سیستم های تجاری، بسیاری از اکوسیستم های اوپن سورس غیر اختصاصی نیز وجود دارند.

# کاربرد اینترنت اشیاء در نگهداری از سالمندان

یکی از کاربردهای مهم خانه هوشمند، کمک به افراد معلول و افراد سالخورده است. این سیستم های خانگی برای رفع ناتوانی های خاص مالک از فناوری کمکی استفاده می کنند. کنترل صدا می تواند به کاربران در محدودیت های دید و تحرک کمک کند در حالی که سیستم های هشدار می توانند به طور مستقیم به کاشت حلزون که توسط کاربران کم شنوا پوشیده می شود متصل شوند. همچنین می توانند به ویژگی های ایمنی اضافی مجهز شوند. این ویژگی ها می تواند شامل حسگرهایی باشد که در مواقع اضطراری پزشکی مانند افتادن یا تشنج نظارت می کنند. فناوری خانه هوشمند که از این طریق استفاده می شود، می تواند آزادی و کیفیت زندگی بیشتری را در اختیار کاربران قرار دهد. اصطلاح "Enterprise IoT" به دستگاه های مورد استفاده در تنظیمات تجاری و سازمانی اطلاق می شود. تخمین زده می شود که تا پایان سال 2019 ، EIoT تعداد 9.1 میلیارد دستگاه را به خود اختصاص دهد.

# کاربرد اینترنت اشیاء در خدمات پزشکی و بهداشتی

اینترنت اشیاء پزشکی (IoMT)، همچنین به آن اینترنت چیزهای سلامت گفته می شود، برنامه IoT برای اهداف پزشکی و بهداشتی، جمع آوری داده ها و تجزیه و تحلیل برای تحقیق و نظارت است. IoMT به عنوان فناوری ایجاد سیستم مراقبت بهداشتی دیجیتالی، اتصال منابع پزشکی موجود و خدمات بهداشتی درمانی، به عنوان "هوشمند درمانی" معرفی شده است.

# کاربرد اینترنت اشیاء در اتوماسیون ساختمان و خانه

از دستگاه های IoT می توان برای نظارت و کنترل سیستم های مکانیکی، برقی و الکترونیکی مورد استفاده در انواع مختلف ساختمان ها (به عنوان مثال عمومی و خصوصی، صنعتی، مؤسسات یا مسکونی) در سیستم های اتوماسیون منزل و اتوماسیون ساختمان استفاده کرد. در این زمینه، سه حوزه اصلی تحت پوشش قرار می گیرد:

* ادغام اینترنت با ساختمان های سیستم های مدیریت انرژی به منظور ایجاد "ساختمان های هوشمند" با انرژی و کارآمد برای IOT.
* ابزار ممکن برای نظارت بر زمان واقعی برای کاهش مصرف انرژی و نظارت بر رفتارهای سرنشینان.
* ادغام دستگاه های هوشمند در محیط ساخته شده و نحوه استفاده آن ها در برنامه های آینده.

# کاربردهای صنعتی اینترنت اشیاء

همچنین با عنوان IIoT شناخته می شود، دستگاه های صنعتی IoT داده های مربوط به تجهیزات متصل، فناوری عملیاتی، مکان ها و افراد را به دست آورده و آنالیز می کنند. IIOT همراه با دستگاه های مانیتورینگ فناوری عملیاتی (OT)، IIOT به تنظیم و نظارت بر سیستم های صنعتی کمک می کند.

# کاربرد اینترنت اشیاء در کشاورزی

چندین کاربرد IoT در کشاورزی وجود دارد مانند جمع آوری داده ها در مورد دما، بارندگی، رطوبت، سرعت باد، آلودگی به آفات و محتوای خاک. از این داده ها می توان برای اتوماسیون تکنیک های کشاورزی، اتخاذ تصمیمات آگاهانه برای بهبود کیفیت و کمیت، به حداقل رساندن ریسک و ضایعات و کاهش تلاش مورد نیاز برای مدیریت محصولات زراعی استفاده کرد.

ما در این پروژه سعی کرده ایم تا سیستمی را طراحی کنیم که پایه تمام نیازهای گفته شده باشد، سیستمی با قابلیت انعطاف پذیری بالا که با استفاده از تکنولوژی های وب و سوکت، امکان کنترل بسیاری از دستگاه های مطرح شده را ارائه می دهد.

# برخی از تکنولوژی های اینترنت اشیا

* AVR
* ARM
* کورتکس (Cortex)
* رزبری پای (Raspberry pi)
* FPGA
* آردوینو (Arduino)

# آردوینو

ایده ساخت آردوینو در سال ۲۰۰۳ میلادی در انستیتو طراحی‌تعاملی ایورئا در کشور ایتالیا شکل گرفت. ایده عبارت بود از ساخت وسیله‌ای ساده و کم‌هزینه برای انجام پروژه‌های دیجیتال دانشجویان، به‌خصوص آن‌هایی که آشنایی چندانی با اصول مهندسی و برنامه‌نویسی ندارند.

آردوینو (به انگلیسی: Arduino) یک پلتفرم سخت‌افزاری و نرم‌افزاری متن باز است. پلتفرم آردوینو شامل یک میکروکنترلری تک‌بردی متن‌باز است که قسمت سخت‌افزار آردوینو را تشکیل می‌دهد. علاوه بر این، پلتفرم آردوینو یک نرم‌افزار آردوینو IDE که به منظور برنامه‌نویسی برای بردهای آردوینو طراحی شده‌است و یک بوت لودر نرم‌افزاری که بر روی میکروکنترلر بارگذاری می‌شود را در بر می‌گیرد. پلتفرم آردوینو به منظور تولید سریع و ساده پروژه‌های سخت‌افزاری تعاملی و ساخت وسایلی که با محیط تعامل داشته باشند طراحی شده‌است.البته بردهای آردوینو اهداف آموزشی را نیز دنبال می‌کنند.

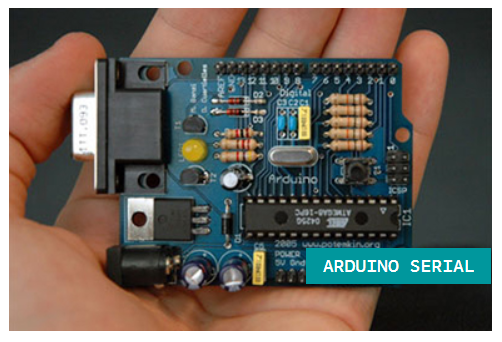
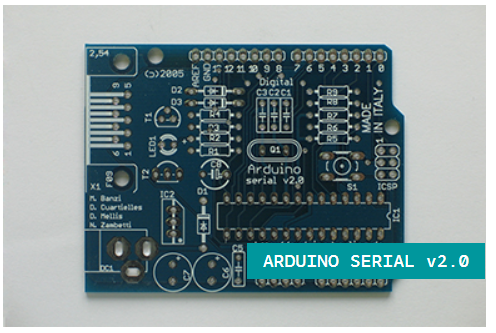
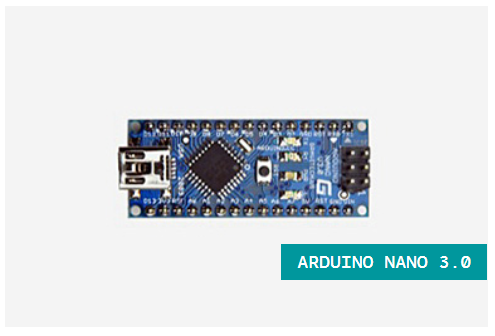
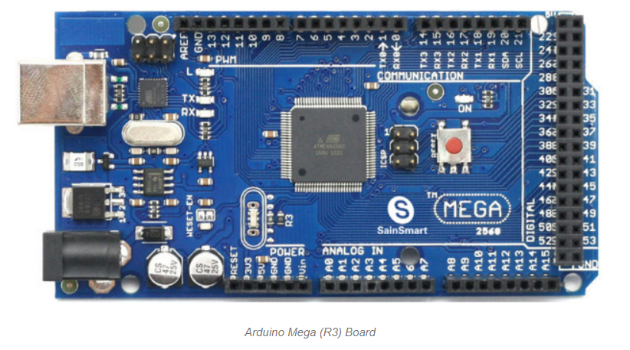
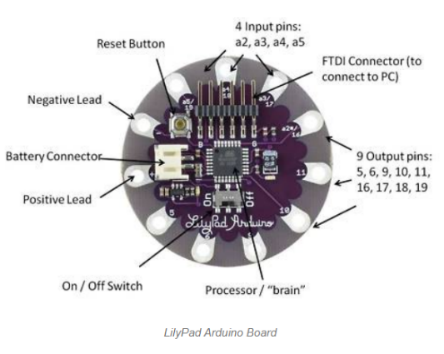
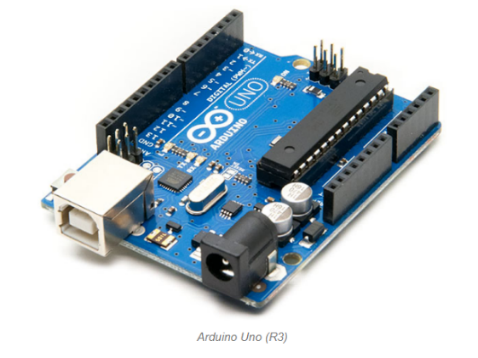
آردوینو می‌تواند جهت طراحی و ساخت سریع و آسان وسایل تعاملی مورد استفاده قرار گیرد. این وسیله در سال ۲۰۰۵ به منظور ایجاد راهی ارزان و ساده برای برنامه‌نویسی اشیایی تعاملی ایجاد شد. آردوینو به همراه یک محیط یکپارچه توسعه نرم‌افزار (IDE) ساده ارائه می‌شود که در رایانه‌های عادی قابل اجرا است که اجازه برنامه‌نویسی به کمک سی یا سی++ را برای آردوینو می‌دهد.

این وسیله می‌تواند پارامترهایی مانند نور محیط، کلیدها یا حتی یک ایمیل را به عنوان ورودی دریافت نماید و بعد از پردازش، خروجی‌هایی مانند روشن کردن یک وسیلهء برقی، تغییر رنگ LEDها یا ارسال یک ایمیل یا نظیر آن را ارائه دهد.

آردوینو انواع متفاوتی دارد، گرچه بسیاری از این مدل ها توسط شرکت اصلی سازنده بازنشسته شده اند اما به خاطر کاربرد زیاد و متنوعشان توسط سازنده های چینی تولید می‌شوند. نام برخی از آن ها عبارت است از:

* آردوینو یون (Arduino YUN)
* آردوینو مگا ADK (Arduino MEGA ADK)
* آردوینو مینی (Arduino MINI)
* آردوینو نانو (Arduino NANO)
* لیلی‌پد
* آردوینو سریال

تصاویر زیر تعدادی از مدل های آردوینو است:



# ماژول‌ها و شیلدهای آردوینو

با استفاده از آردوینو و به کمک ماژول‌ها و شیلدهای متنوعی که برای آردوینو ساخته شده می‌توان پروژه‌های مختلفی مانند اینترنت اشیا، خانه‌های هوشمند و محصولات تعاملی را پیاده‌سازی کرد.

بردهای آردوینو را بدون نیاز به پروگرامر می‌توان از طریق کابل USB و محیط یکپارچه توسعه نرم‌افزار متن‌باز آردوینو پروگرام کرد.

برای استفاده در کنار بوردهای آردوینو ماژول های زیادی وجود دارند تعدادی از آن ها که در اینترنت اشیا استفاده بیشتری دارند عبارت اند از:

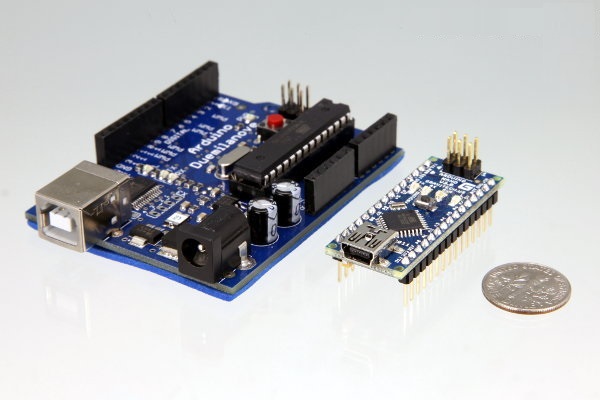
* Arduino buzzer module
* Arduino fire module
* Arduino WiFi module
* Arduino Relay module
* Arduino tracking module

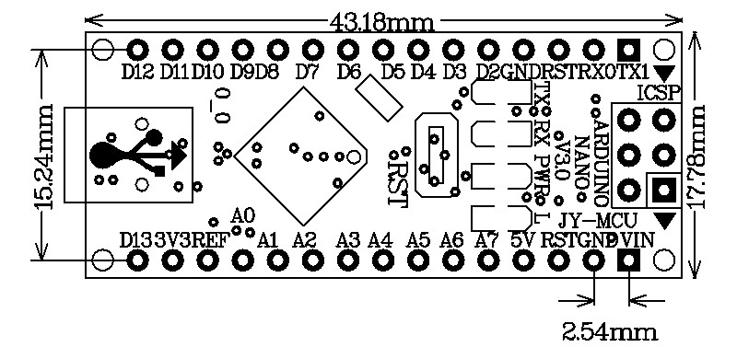


# آردوینو nano

این برد با میکروکنترلر ATMEGA328P دقیقا مشابه آردوینو UNO کارمی کند . از لحاظ کارایی و تعداد پین ها هیچ تفاوتی با آردوینو UNO ندارد. مزیت اصلی این برد ابعاد بسیار کوچک و قیمت مناسب آن است که به راحتی می توانید از آن به طور مستقیم در پروژه های خود مورد استفاده قرار دهید.

از آردوینو نانو در بسیاری از پروژه ها که محدویت فضا دارند من جمله ربات ها می توان استفاده کرد.





# مشخصات برد آردوینو nano

میکروکنترلر: ATmega328

ولتاژ کاری: 5 ولت

ولتاژ ورودی پیشنهادی: 7 تا 12 ولت

محدوده ولتاژ ورودی: 6 تا 20 ولت

تعداد پینهای I/O دیجیتال: 14

تعداد پینهای آنالوگ ورودی: 8

جریان ورودی برای هر پین: 40 میلی آمپر

حافظه فلش: 32 کیلوبایت

حافظه SRAM: کیلوبایت

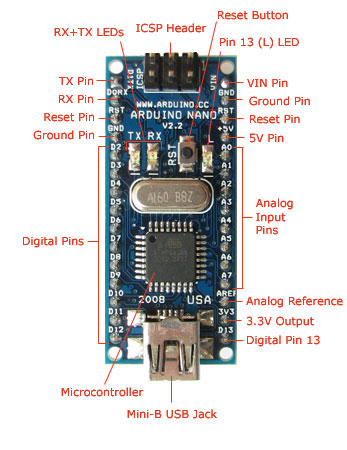
حافظه EEPROM: یک کیلوبایت

فرکانس کاری: 16 مگاهرتز

ابعاد: 0.73 در 1.7 اینچ

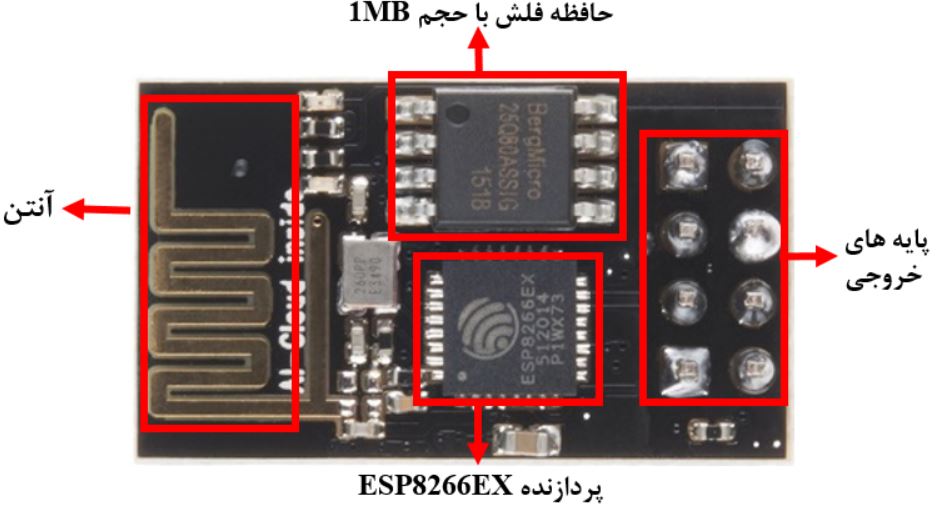
آردوینو نانو به راحتی با استفاده از کابل MINI USB در محیط نرم افزار آردوینو پروگرام می شود.

ترتیب پایه های آن به صورت زیر است:



# ماژول WiFi

ESP8266 میکرو کنترلری ساخت شرکت Espressif Systems که به دلیل داشتن قابلیت انجام فعالیت های مربوط به WiFi بیشتر به عنوان ماژول WiFi شناخته شده است. لیکن به خاطر میکروکنترلر بودن آن، می‌توان آن را برای پردازش اطلاعات استفاده کرد.



# کاربردهای ماژول ESP8266

* اتوماسیون خانگی
* برنامه های پزشکی
* کنترل و مانیتورینگ بی سیم
* دوربین های IP دار
* سیستم های امنیتی

# انواع ماژول ESP8266

این ماژول انواع مختلفی دارد که از ESP8266-01 تا ESP8266-12 نام گذاری شده اند. تمام این مدل ها از پردازنده ای یکسان استفاده می‌کنند و تفاوت آن ها در تعداد پایه های در دسترس و شکل ظاهری آن‌هاست.

تصویر برخی از این مدل ها در ادامه آمده است.



در این پروژه از مدل ESP8266-01 که دسترسی به آن راحت تر است استفاده می‌کنیم.

# حالت های ماژول ESP8266

این ماژول می‌تواند به سه حالت عمل کند:

* در نقش Access Point
* در نقش Station
* حالت ترکیبی Station و Access Point

# در نقش Access Point



# در نقش Station



# حالت ترکیبی Station و Access Point

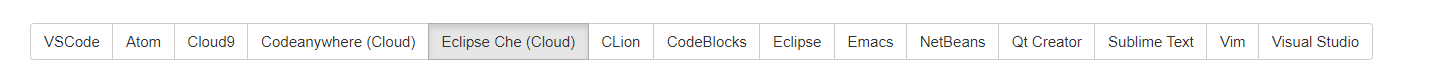


# Platform IO

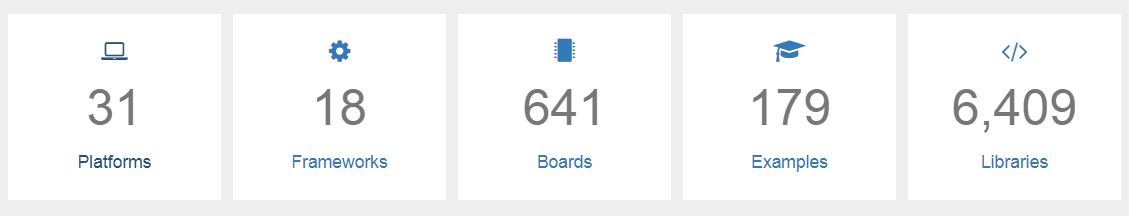
در کنار IDE شرکت آردوینو، می‌توان از محیط های دیگری برای برنامه نویسی آن استفاده کرد.

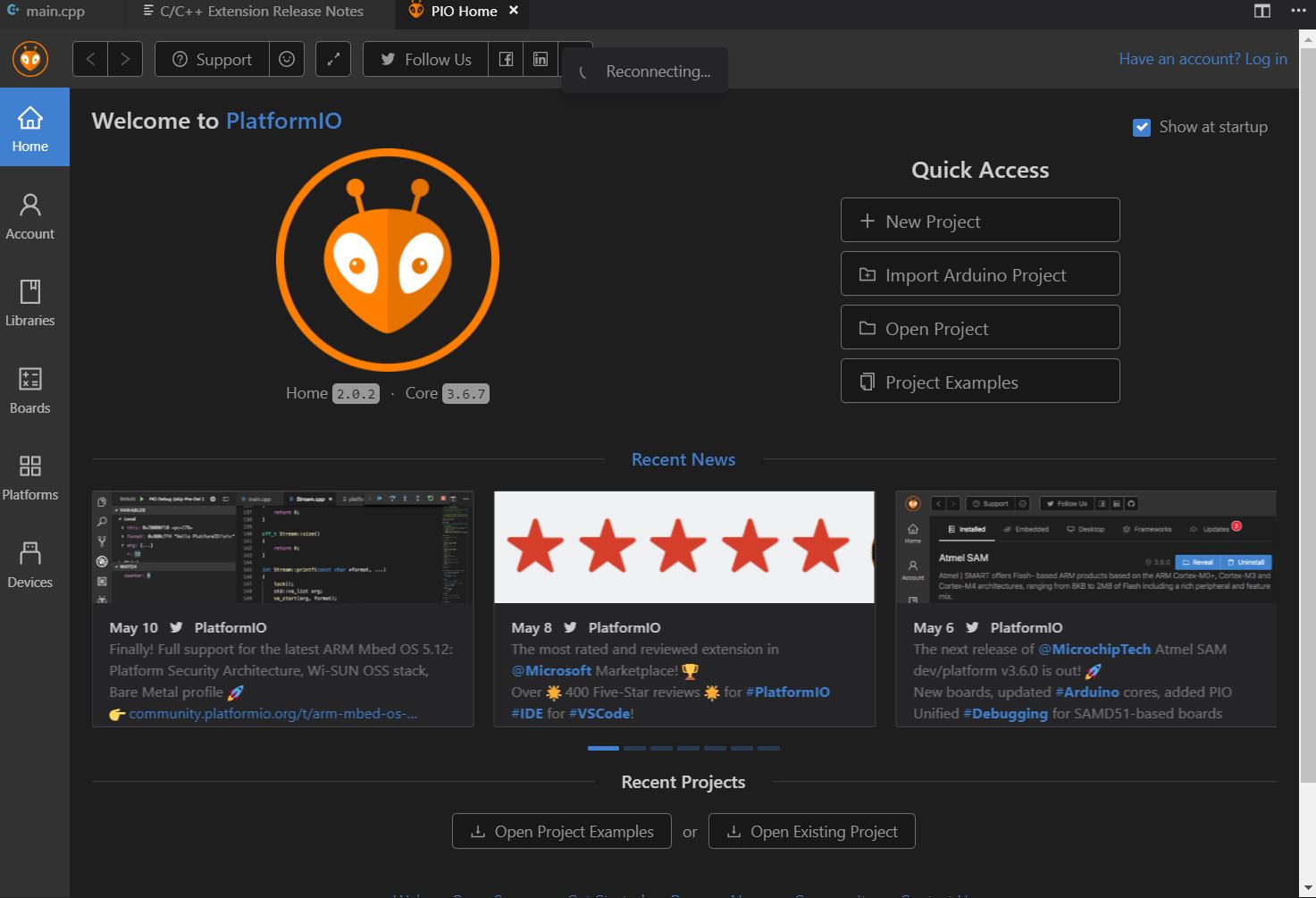
محیط مورد استفاده در این پروژه، PlarformIO است که برای توسعه اینترنت اشیاء مناسب است.

این محیط با فراهم کردن امکانات بهتر برای دیباگ و توسعه و نگه داری کدهای بیشتر، محیط مناسب تری از Arduino IDE می‌باشد و به صورت افزونه برای اضافه کردن در vscode، atom و .. وجود دارد.



در این محیط به راحتی می‌توان بورد ، فریم ورک کتابخانه مورد نظر را پیدا و استفاده کرد.





# برنامه نویسی سرور

برای قسمت سرور از پلتفرم Node.js استفاده می کنیم.

# معرفی نود جی اس (Node.js)

Node.js یک محیط اجرایی اوپن سورس و چندسکویی برای جاوااسکریپت است که کد جاوا اسکریپت را در خارج از یک مرورگر اجرا می کند. Node.js به توسعه دهندگان اجازه می دهد تا از JavaScript برای نوشتن ابزارهای خط فرمان و اسکریپت نویسی سمت سرور استفاده کنند. در نتیجه، Node.js با یکپارچه سازی توسعه برنامه وب حول یک زبان برنامه نویسی واحد، به جای زبان های مختلف برای اسکریپت های سمت سرور و کلاینت به عنوان یک نمونه جاوااسکریپت در همه جا نشان داده می شود.

گرچه js. فرمت استاندارد فایل برای کد JavaScript است، اما نام "Node.js" در این زمینه به فایل خاصی اشاره نمی کند و صرفاً نام محصول است. Node.js دارای معماری مبتنی بر رویداد است که توانایی ورودی/خروجی ناهمگام یا غیر همزمان را دارد. این گزینه های طراحی با هدف بهینه سازی توان و مقیاس پذیری در برنامه های وب با بسیاری از عملیات ورودی/خروجی و همچنین برای برنامه های وب در Real-time (به عنوان مثال، برنامه های ارتباطی در Real-time و بازی های مرورگر) انجام می شود. پروژه توسعه توزیع شده Node.js که توسط بنیاد Node.js اداره می شود، توسط برنامه پروژه های همکاری بنیاد لینوکس تسهیل می شود.

* فایل های نود جی اس شامل task هایی هستند که در برخی از موارد خاص اجرا می شوند.
* یک رویداد معمولی سعی در دسترسی به پورت روی سرور دارد.
* فایل های Node.js باید قبل از هرگونه نتیجه ای، روی سرور آغاز شوند.
* فایل های Node.js دارای پسوند ".js" هستند.

# کاربردهای نود جی اس (Node.js)

* طراحی وب
* طراحی وب سرویس
* در اینترنت اشیا (IoT)
* اپلیکیشن های مبتنی بر json
* اپلیکیشن های تک صفحه ای
* Data Intensive Real-time Applications (DIRT)
* اپلیکیشن های جریان داده
* اپلیکیشن های I/O bound

# ویژگی های نود جی اس (Node.js)

**غیرهمزمانی و مبتنی بر رویداد:** تمام API های کتابخانه Node.js غیرهمزمان هستند. اساسا به این معنی است که یک سرور مبتنی بر Node.js هرگز منتظر API برای بازگشت داده ها نیست. سرور بعد از فراخوانی آن به API بعد حرکت می کند و مکانیزم اعلان رویدادهای Node.js به سرور کمک می کند تا از فراخوانی API پیشین پاسخی دریافت کند.

**تک Threaded اما بسیار مقیاس پذیر:**Node.js از یک مدل Threaded با حلقه رویداد استفاده می کند. مکانیزم رویداد به سرور کمک می کند که به یک روش غیر مسدود کننده پاسخ دهد و سرور را بسیار مقیاس پذیر می کند، به نحوی که در مقایسه با سرورهای سنتی که threads محدودی را برای رسیدگی به درخواست ایجاد می کنند،Node.js از یک برنامه تک threaded استفاده می کند و همان برنامه می تواند برای سرویس تعداد زیادی درخواست فراهم کند که از سرویس های سنتی مثل Apache HTTP Server بیشتر می باشد.

**بدون Buffering:** اپلیکیشن های Node.js هرگز هیچ داده ای را بافر نمی کنند. این برنامه ها به سادگی داده ها را درون chunks قرار می دهند.

# RxJS

ReactiveX یک ابزار یا کتابخانه برای Asynchronous Programming یا برنامه نویسی غیرهمزمان با استفاده از Observable ها می باشد. این ابزار برای زبانهای مختلف پیاده سازی شده است و RxJS پیاده سازی این ابزار در زبان Javascript می باشد.

RxJS 7 یک بازنویسی از Reactive-Extensions / RxJS و آخرین نسخه RxJS است.

در این بازنویسی عملکرد و قدرت اشکال یابی افزایش یافته.

# Express

اکسپرس‌ جی‌ اس (Express js)، یا بطور ساده، اکسپرس (Express) یک فریم‌ورک انعطاف پذیر وب برای نود جی‌ اس است که مجموعه‌ ای از ویژگی ها همچون APIهای ساده در اختیار قرار می‌دهد تا در ساختن بخش مدیریت (بک-اند یا back-end) در یک وبسایت یا اپلیکین موبایل از آنها استفاده کنیم. اکسپرس متن باز است و درحال حاضر توسط موسسه نود جی‌ اس پشتیبانی می‌شود. اکسپرس (Express js) همچنین میان‌افزارها (middlewares) و تعاریف مسیرنویسی (Routing) پایه را برای ما فراهم می‌سازد.

* روتینگ یا مسیرنویسی را بسیار ساده می‌کند.
* به دلیل اینکه برپایه‌ی نود جی‌ اس بنا شده، بسیار سریع است.
* قابل اجرا برای تولید سریع برنامه های کاربردی.

# socket.io

چارچوب Socket.IO ارتباطات مبتنی بر رویداد دو طرفه و real time را امکان پذیر می کند. و شامل دو قسمت است:

* سرور Node.js
* کتابخانه برای کلاینت Node.js یا مرورگر

پیاده سازی این فریمورک در زبان های دیگر نیز انجام شده:

* جاوا
* C++
* سوییفت

همچنین اتصال و استفاده از آن در حضور پراکسی و firewall و برنامه های آنتی ویروس روی کلاینت نیز بدون مشکل قابل انجام است.

# نحوه کار برنامه سرور

برنامه نویسی سرور این پروژه با استفاده از node.js انجام شده. و برای ارتباط با ماژول و همچنین client سرور سوکت ایجاد شده.



شماره پورت سرور درصورت وجود از محیط پردازه خوانده می شود و در غیر اینصورت مقدار پیشفرض 3303 قرار می گیرد.

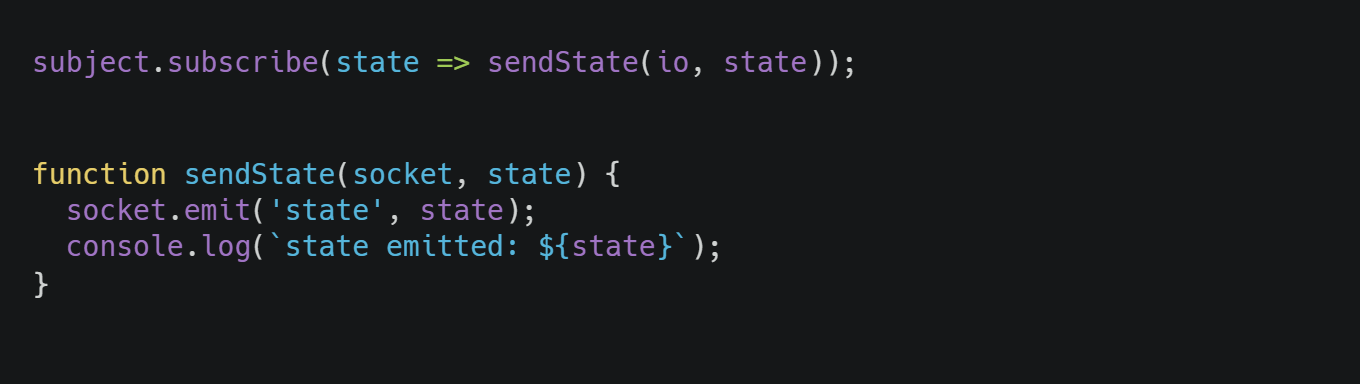
با استفاده از کتابخانه express یک listener و با استفاده از کتابخانه های http و socket.io یک سرور سوکت ساخته شده.



با استفاده از RxJs یک wrapper برای نگه داری وضعیت روشن و خاموش سوئیچ، ایجاد امکان تغییر و گوش دادن به این تغییر ایجاد شده است.

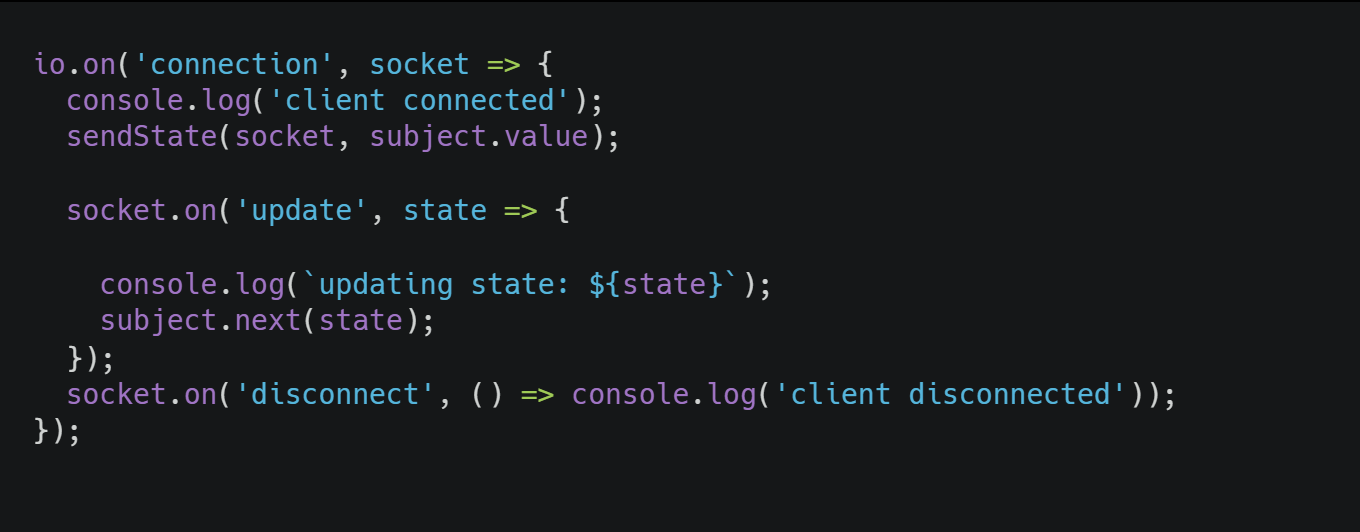


برای اطمینان از اجرای صحیح برنامه، با وارد کردن آدرس (URL) سرور در یک مرورگر، وضعیت فعلی نمایش داده می شود.



به تغییرات وضعیت گوش داده می‌شود تا با هر تغییر وضعیت، وضعیت جدید به تمام کلاینت های سرور-سوکت فرستاده شود.

تابع sendState نیز برای ارسال وضعیت به سوکت مورد نظر نوشته شده.



به سرور سوکت گوش داده می‌شود تا اگر کلاینت جدید اضافه شود با خبر شویم.

اگر کلاینت جدیدی متصل شد ابتدا وضعیت (state) را به آن می‌فرستد. سپس به همان کلاینت گوش می‌دهد تا اگر رویدادی (event) از سمت آن اتفاق بیفتد وضعیت (state) را به روز رسانی کند.

و در صورت قطع اتصال گزارش (log) قطع اتصال بدهد.



در این خط سرور با پورت مشخص شده شروع به کار می کند.

پس از طراحی سرور میتوان کدهای نوشته شده را در Heroku که یک ارائه دهنده رایگان سرور های اختصاصی است آپلود کرد تا آن را اجرا و استفاده کنیم. آدرس سرور در این پلتفرم به صورت زیر است:

<https://iot-switch-server.herokuapp.com/>

# برنامه نویسی پنل کنترل کاربری اندروید

قسمت کلاینت پروژه برای اندروید به صورت native و با زبان کاتلین نوشته شده است.

# کاتلین (Kotlin)

کاتلین یک زبان برنامه نویسی چند سکویی، استاتیک و همه منظوره با استنتاج نوع است. کاتلین توسط JetBrains طراحی شده و این طراحی به گونه ای انجام شده که به طور کامل با جاوا همکاری کند. نسخه JVM کتابخانه استاندارد آن به کتابخانه کلاس جاوا بستگی دارد، اما استنتاج نوع باعث می شود که سینتکس آن مختصر تر باشد.

سینتکس Kotlin ممکن است دقیقا مشابه با JAVA نباشد، با این حال، در داخل Kotlin کتابخانه کلاس Java موجود است تا نتایج فوق العاده برای برنامه نویسان تولید کند. قابلیت تعامل، ایمنی کد و وضوح را برای توسعه دهندگان در سراسر جهان فراهم می کند.

کاتلین به طور رسمی توسط Google برای توسعه تلفن همراه در اندروید پشتیبانی می شود. از زمان انتشار اندروید استودیو 3.0 در اکتبر 2017، کاتلین به عنوان جایگزینی برای کامپایلر استاندارد جاوا درج شده است. کامپایلر اندروید کاتلین به کاربر اجازه می دهد تا از بین کد های سازگار با جاوا 6 یا جاوا 8 استفاده کند.کاتلین از مه 2019 زبان اصلی مورد علاقه Google برای توسعه برنامه اندروید است. در سال 2017، Google اعلام کرد Kotlin زبان رسمی توسعه اندروید است. کاتلین یک زبان برنامه نویسی اوپن سورس است که ویژگی های برنامه نویسی شی گرا و تابعی را در یک پلتفرم منحصر به فرد ترکیب می کند. همچنین از طریق بنیاد کاتلین توسط JetBrains و Google حمایت مالی می شود.

# ویژگی های کاتلین

* کاهش خطا در زمان اجرا
* هزینه پایین
* دارای Syntax مشابه جاوا
* کاتلین بر اساس JVM و یک زبان تابعی است که کدنویسی را مختصر می کند
* دارای عکلکرد بهتر و زمان اجرای کمتر
* ساده سازی ساخت اپلیکیشن های تعاملی

در این بخش تنها کتابخانه مورد استفاده، پیاده سازی قسمت کلاینت socket.io به زبان جاوا است.

با استفاده از کاتلین و کتابخانه socketio.client دو کار انجام می‌گیرد:

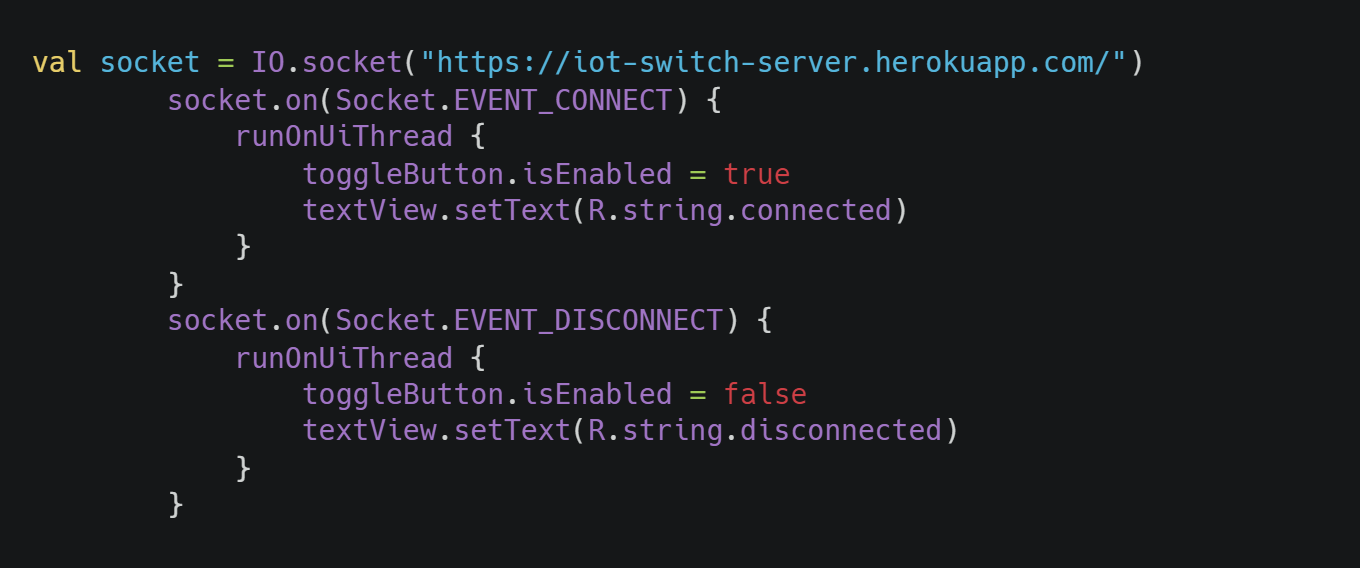
1. نظارت (watch) بر وضعیت (state) ازطریق سرور
2. تغییر(update) وضعیت به روشن یا خاموش

# نحوه کار کد اندروید



پس از ایمپورت کتابخانه های مورد نیاز، در هنگام اجرای برنامه، دو تا از اجزای ظاهری که یکی متنی برای نمایش وضعیت اتصال و دیگری دکمه ای برای نمایش و کنترل وضعیت سوئیچ است را در متغیرهای جداگانه ذخیره میکنیم.

در ابتدا دکمه ی کنترل حالت (state) غیر فعال می‌شود تا در صورت عدم برقراری اتصال به سرور پیام تغییر حالت ارسال نشود.



برنامه اقدام به اتصال به سرور می‌کند و منتظر برقراری کامل اتصال می‌ماند.

در صورت اتصال و گرفتن پیام وضعیت از سرور، وضعیت در برنامه نمایش داده می‌شود و دکمه برای تغییر وضعیت از سمت کاربر فعال می‌شود.

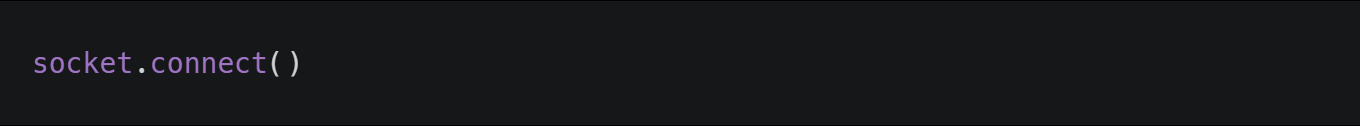
در صورت قطع اتصال بین سرور و برنامه دکمه نیز غیر فعال می‌شود.



برنامه به وضعیت سرور گوش می‌دهد تا با عوض شدن state از سمت سرور در رابط کاربری وضعیت جدید نمایش داده شود.

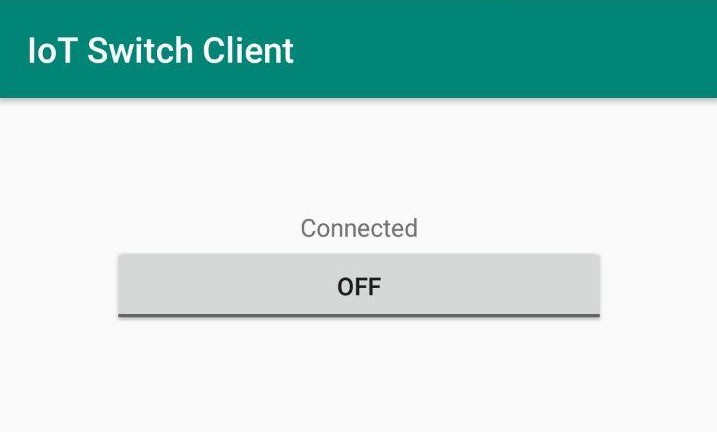


به فرمان تغییر وضعیت از سمت کاربر در دکمه گوش داده می‌شود و در صورت انجام این فرمان به سرور دستور تغییر حالت داده می‌شود.



اتصال به سرور-سوکت آغاز می‌شود.

# تصویر نهایی برنامه

****

# نصب برنامه

فایل برنامه از آدرس زیر قابل دریافت و نصب است:

<https://github.com/SadraSamadi/iot-switch/blob/master/IoT_Switch_Client_v1.0.0.apk>

# برنامه نویسی برد آردوینو

برای برنامه نویسی این بخش از PlatformIO و کتابخانه های موجود در آن استفاده میکنیم.

# ArduinoJson

یک کتابخانه C++ است که برای کار با فرمت JSON در Arduino و اینترنت اشیا کاربرد دارد.

# نحوه کار برنامه برد آردوینو

همانطور که گفته شد، ماژول wifi به صورت جداگانه قابلیت برنامه ریزی را داراست، اما به دلیل اینکه تعداد پورت های ورودی و خروجی آن محدود است ما از برد آردوینو استفاده می کنیم تا فقط دستورات را از ماژول توسط نوع ارتباطی UART گرفته و کنترل خروجی را به برد آردوینو واگذار کنیم.

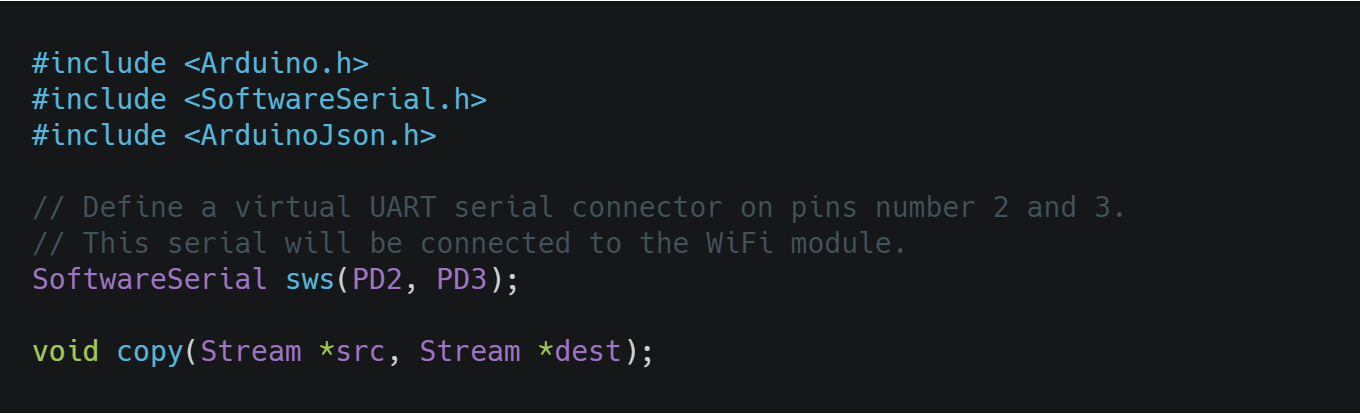
همچنین برای پروگرم کردن ماژول ما نیاز به یک مبدل UART به USB داریم که برد آردوینو این قابلیت را هم برای ما فراهم می کند.

برای این منظور دو حالت برای آردوینو در نظر گرفته ایم که توسط یک سوئیچ کنترل می شود.

حالت اول برد را به یک پل ارتباطی تبدیل می کند که یک سمت آن به پورت USB کامپیوتر و سمت دیگر از طریق UARTبه ماژول متصل میشود. به این صورت ماژول توسط کامپیوتر شناخته و پروگرم می شود.

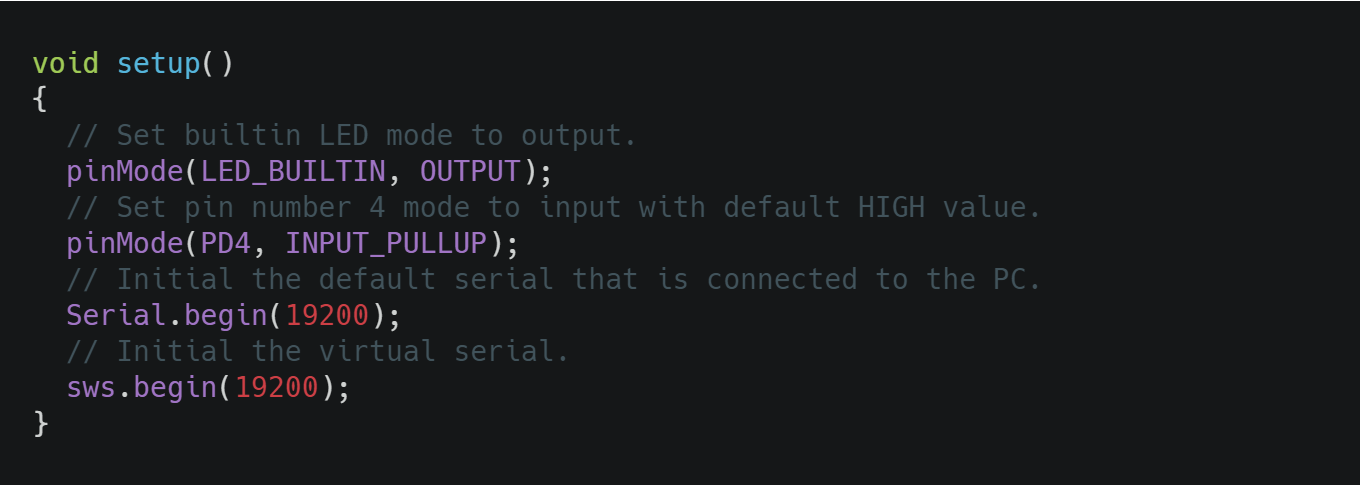
حالت دوم که پس از اتمام پروگرم کردن استفاده می شود، فقط از طریق UART دستورات را از ماژول wifi گرفته و خروجی مورد نظر را اعمال می کند.

# نحوه کار کد برد آردوینو

****

پس از وارد کردن کتابخانه های مورد نیاز، ابتدا یک رابط سریال نرم افزاری روی پورت های 2 و 3 آردوینو تعریف می کنیم تا به ماژول WiFi متصل شود. رابط سریال پیشفرض سخت افزاری مستقیم به کامپیوتر متصل می شود.

پس از آن یک تابع برای کپی داده تعریف شده که در ادامه توضیح داده خواهد شد.



این متد در ابتدای راه اندازی برد اجرا می شود.

ابتدا چراغ داخلی برد را به صورت خروجی تعریف می کنیم.

سپس پایه 4ام برد را با یک مقاومت بالاکش به صورت ورودی تعریف می کنیم. این پایه در ادامه برای تعیین عملکرد برد استفاده می شود.

در نهایت سریال های تعریف شده با سرعت های مشخص شده شروع به کار می کنند.



تابع loop پس از setup به طور مداوم تا زمانی که برد روشن است فراخوانی می شود.

دراینجا بسته به عملکردی که مورد نیاز است اگر پایه 4 فعال باشد، برد آردوینو داده ها را از ماژول WiFi که به صورت json است دریافت کرده، تحلیل می کند و عملیات موردنیاز را انجام می دهد.

اگر پایه غیرفعال باشد، برد تنها به صورت واسط عمل کرده و داده هارا بین کامپیوتر و ماژول منتقل می کند.



تابع copy داده ها را از یک منبع خوانده و به مقصد مورد نظر می فرستد.

با نصب افزونه PlatformIO در ویرایشگر Visual Code میتوان به راحتی کدهای نوشته شده را ویرایش، کامپایل و پروگرم کرد.

# برنامه نویسی ماژول WiFi

برنامه نویسی این بخش به عنوان مهم ترین قسمت پروژه شناخته می شود. زیرا چالش اصلی ما متصل کردن ماژول به یک نقطه دسترسی WiFi و انتقال داده از طریق آن بود. اما به لطف پروژه PlatformIO و کتابخانه های مناسبی که در آن وجود دارد این قسمت با موفقیت به پایان رسید.

یکی از ویژگی های مهم این پلتفرم استفاده از فریموورک آردوینو در ماژول WiFi است.

برای کار کردن با ماژول اولین مرحله اتصال آن به یک نقطه دسترسی است. بدین منظور ما از کتابخانه WifiManager استفاده کردیم که در هنگام راه اندازی، ماژول را به حالت نقطه دسترسی برده و روی یک IP خاص یک صفحه web برای نمایش نقاط دسترسی موجود و اتصال به یک از آن ها را فراهم می کند.

پس از آن، ماژول از طریق سوکت و کتابخانه SocketIoClient به سرور متصل می شود و سپس منتظر دریافت دستورات از سرور می ماند تا به محض رسیدن، آن ها را از طریق UART به برد آردوینو منتقل کرده تا به خروجی برسند.

# ArduinoJson

یک کتابخانه C++ است که برای کار با فرمت JSON در Arduino و اینترنت اشیا کاربرد دارد.

# Ticker

این کتابخانه برای آردوینو طراحی شده و اجازه می دهد توابع مورد نیاز را به طور مداوم با تاخیر مشخص فراخوانی کرد.

# WebSockets

کتابخانه آردوینو برای کار با پروتوکول وب سوکت (ws) در ماژول WiFi.

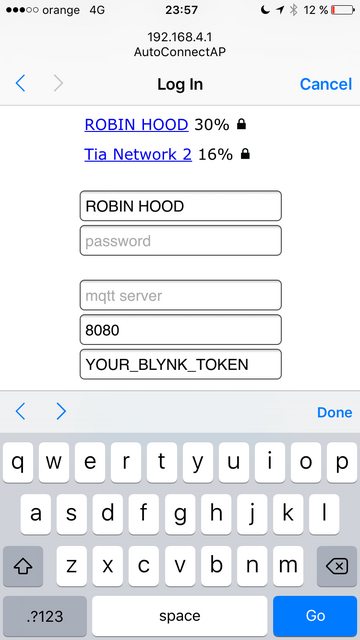
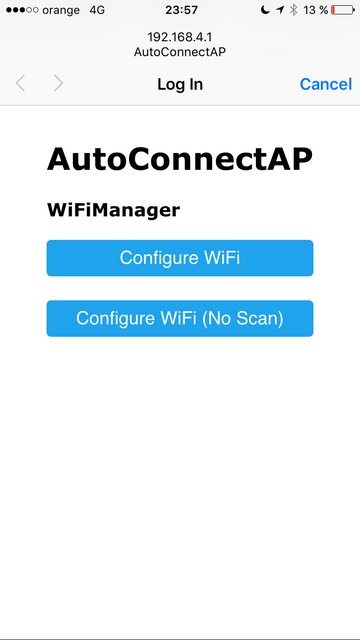
# SocketIOClient

پیاده سازی قسمت کلاینت کتابخانه socket.io که برای آردوینو و ماژول ESP8266 با زبان c++ نوشته شده است.

# WiFiManager

این کتابخانه با استفاده از پیاده سازی آردوینو ماژول ESP8266 امکان ایجاد یک پورتال مبتنی بر وب برای پیکربندی آن را فراهم می کند.

برای دسترسی به این پورتال ابتدا باید از یک دستگاه دیگر (لپتاپ، موبایل و...) که دارای WiFi است به ماژول متصل شده و سپس از طریق مرورگر به آدرس 192.168.4.1 برویم که با صفحات زیر مواجه خواهیم شد.



# نحوه کار کد ماژول WiFi



ابتدا کتابخانه های مورد نیاز را وارد می کنیم.

دو متغیر ثابت برای شماره پورت های ورودی/خروجی تعریف می کنیم.

توابع مورد استفاده را معرفی می کنیم.

یک Ticker با دوره 500 میلی ثانیه ای تعریف می کنیم.

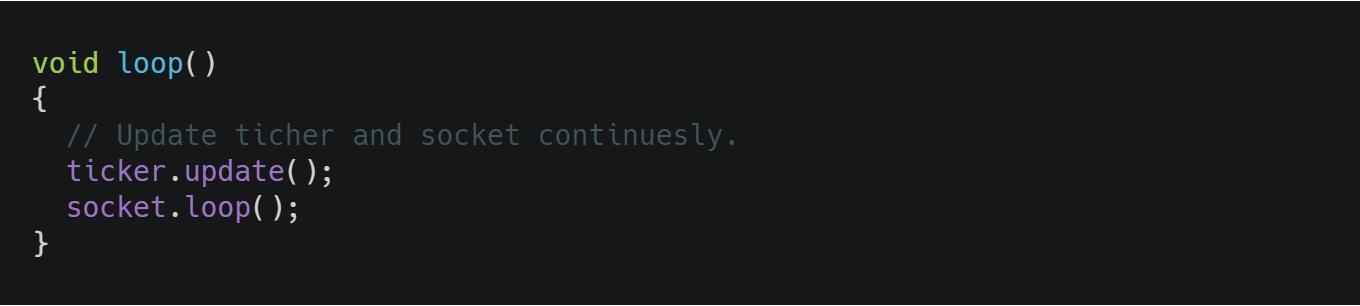
و یک نمونه از کلاینت socket.io میسازیم.



در قسمت setup ابتدا پورت ها را به عنوان خروجی تعریف می کنیم که به دو عدد LED متصل می شوند، یکی برای وضعیت اتصال که در ابتدا یکسره روشن و یکی برای وضعیت روشن و خاموش سوئیچ که در ابتدا خاموش است. سپس رابط سریال را با سرعت مشخص شده شروع می کنیم.

یک نمونه از کلاس WiFiManager می سازیم، اطلاعات قبلی را ریست می کنیم و توسط تابع autoConnect پورتال اشاره شده را راه اندازی کرده و تا پیدا کردن یک مودم WiFi و اتصال به آن در همین خط منتظر می مانیم. پس از اتصال IP اختصاص داده شده به ماژول را به سریال میفرستیم.

در نهایت Ticker و سوکت را آغاز کرده و منتظر واقعه (event) تغییر وضعیت از سوی سرور می مانیم.



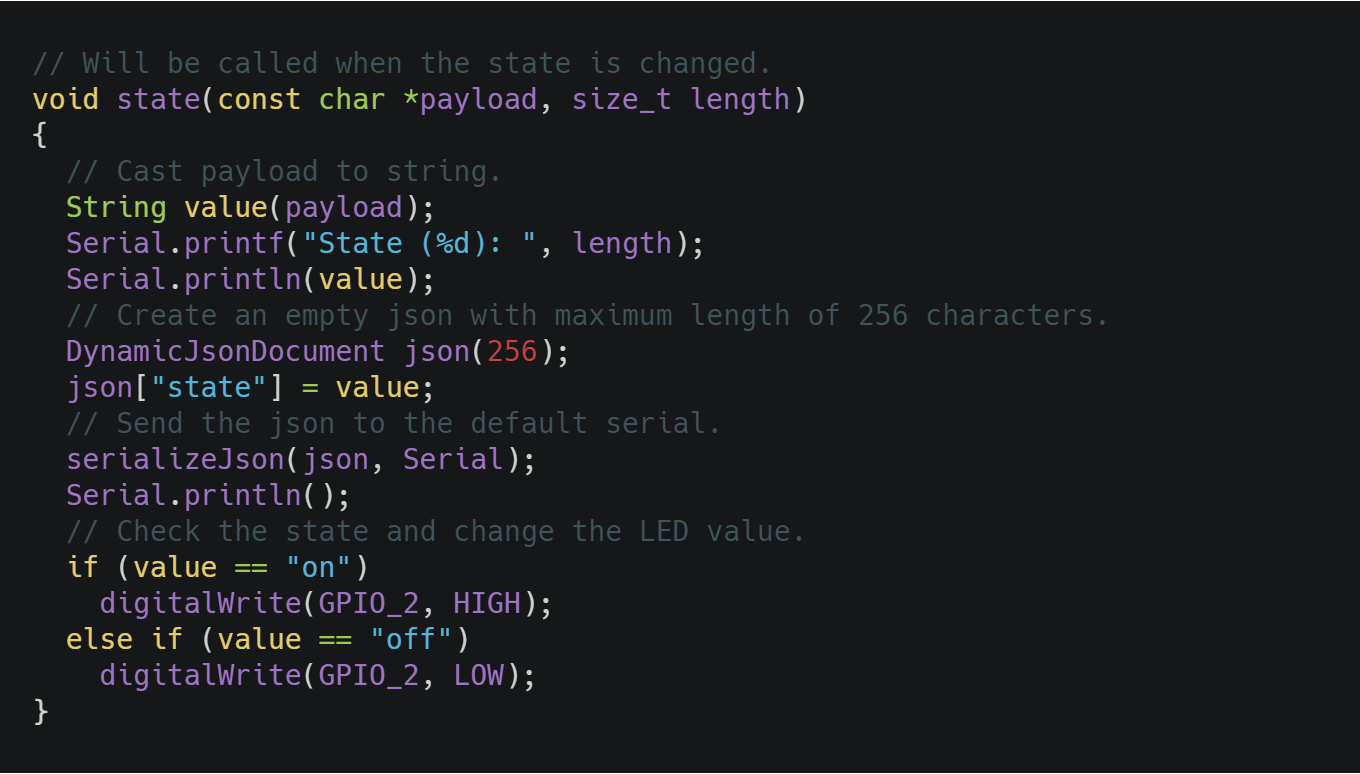
در قسمت loop سوکت و Ticker را به طور مرتب بروزرسانی می کنیم.



در تابع tick وضعیت پایه صفرم ماژول را بررسی کرده، اگر غیرفعال بود آن را فعال می کنیم و بالعکس.

این تابع توسط Ticker کنترل شده و هر 500 میلی ثانیه فراخوانی می شود.

چشمک زدن LED متصل به این پایه نشان می دهد که ماژول با موفقیت به سرور سوکت متصل شده است.



این تابع زمانی فراخوانی می شود که وضعیت سوئیچ از سمت سرور تغییر کند.

برای یکپارچه سازی این تغییرات با فرمت json منتقل می شوند.

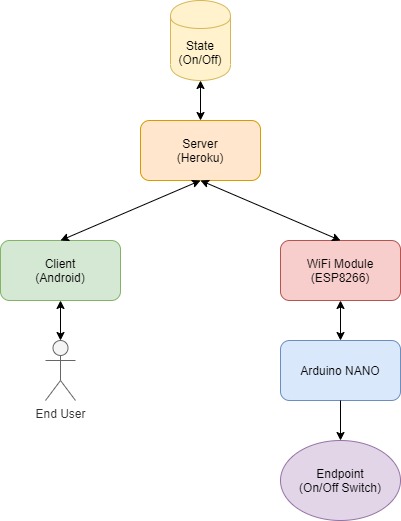
پس از دریافت تغییرات آن را تحلیل کرده و به رابط سریال برای آردوینو میفرستیم.

LED مربوطه را نیز متناسب با وضعیت تغییر می دهیم.

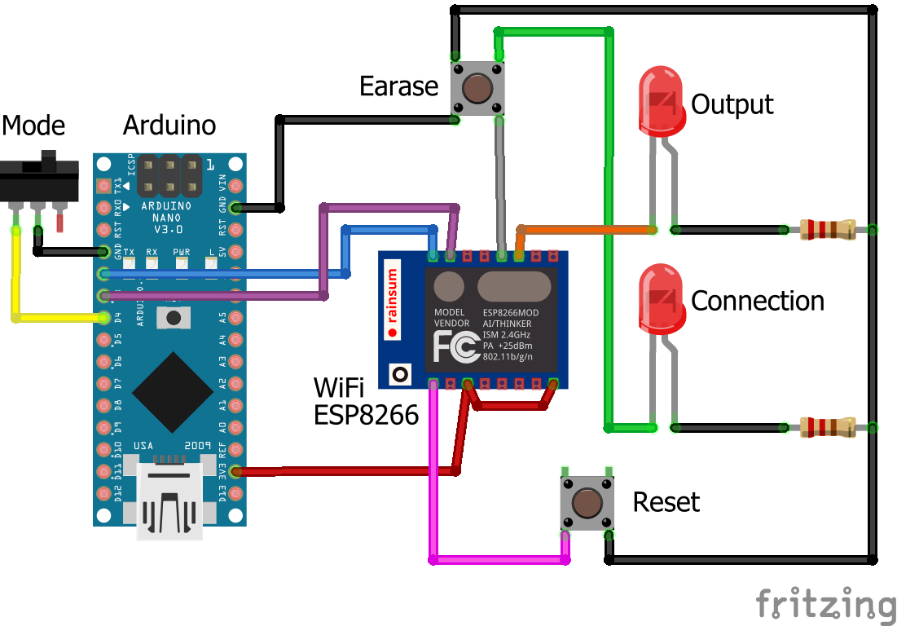
تمامی کدهای اشاره شده در گیتهاب به آدرس زیر قابل دسترس هستند:

<https://github.com/SadraSamadi/iot-switch>

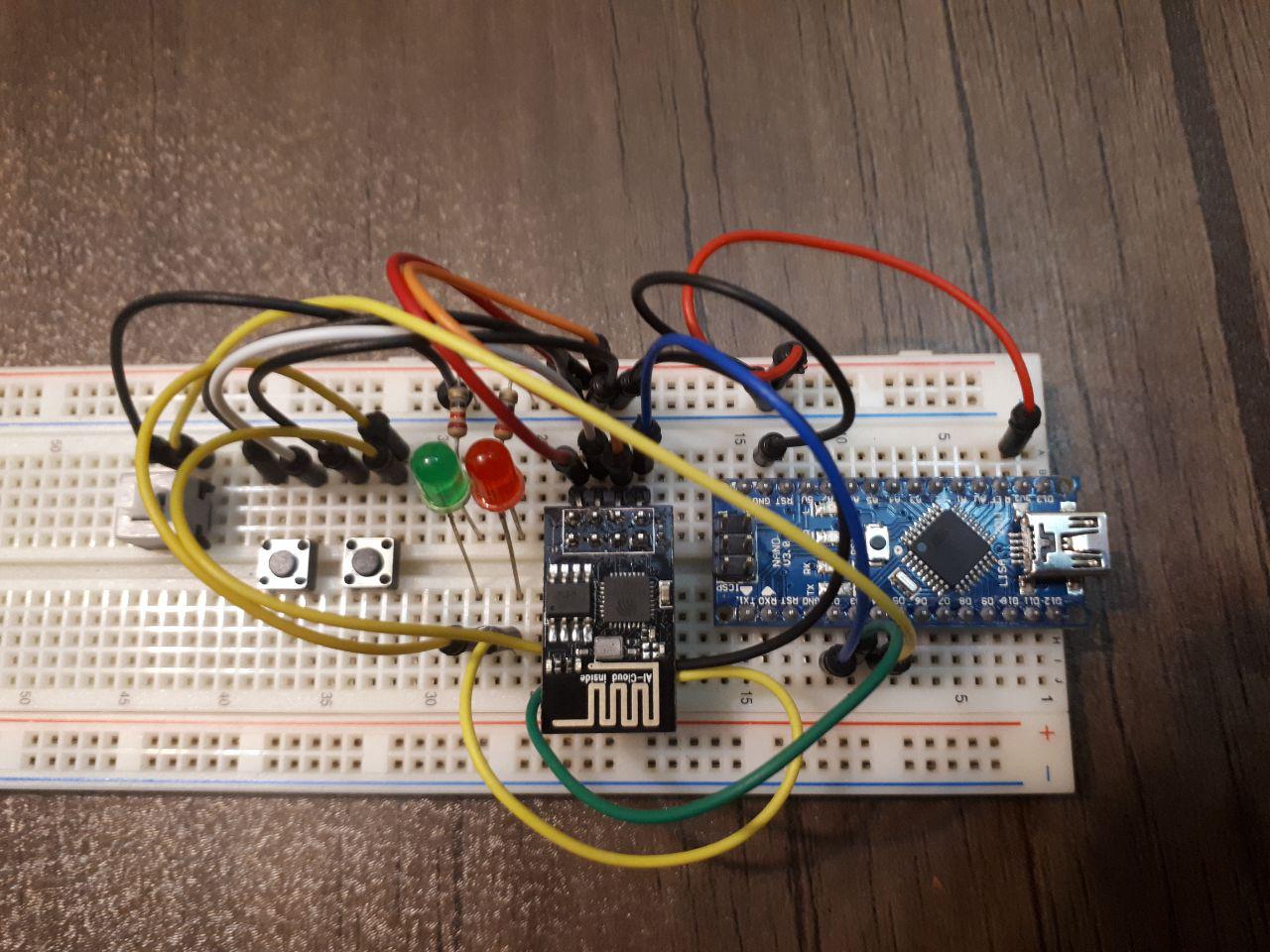
# شمای کلی پروژه



# شمای کلی برد سخت افزاری



# تصویر نهایی برد سخت فزاری



# منابع

<https://nodejs.org/en/>

<https://socket.io/>

<https://expressjs.com/>

<https://rxjs.dev/>

<https://developer.android.com/>

<https://kotlinlang.org/>

<https://github.com/socketio/socket.io-client-java>

<https://www.arduino.cc/en/Guide/ArduinoNano>

<https://www.espressif.com/en/products/hardware/esp8266ex/overview>

<https://platformio.org/>

<https://github.com/esp8266/Arduino>

<https://arduinojson.org/>

<https://github.com/sstaub/Ticker>

<https://github.com/Links2004/arduinoWebSockets>

<https://github.com/timum-viw/socket.io-client>

<https://github.com/tzapu/WiFiManager>