**پیاده سازی اتاق هوشمند**

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی‌کامپیوتر

**پروژه کارشناسی**



محمد امین عیسایی (۹۴۱۰۹۶۲۴)

فاطمه ابوطالبی (۹۴۱۰۵۲۷۱)

سال تحصیلی ۹۷-۱۳۹۸

# فهرست مطالب

[چکیده 3](#_Toc529789546)

[مقدمه 4](#_Toc529789547)

[رزبری‌پای 5](#_Toc529789548)

[ESP8266 6](#_Toc529789549)

[صفحه نمایش 7](#_Toc529789550)

[سنسورها 8](#_Toc529789551)

[کارتخوان 8](#_Toc529789552)

[مادون قرمز 9](#_Toc529789553)

[سنجش دما و رطوبت 10](#_Toc529789554)

[تشخیص دود 11](#_Toc529789555)

[موتور 12](#_Toc529789556)

[بردها 13](#_Toc529789557)

[Programmer 13](#_Toc529789558)

[تنظیم دما 14](#_Toc529789559)

[روشنایی 15](#_Toc529789560)

[دما و تشخیص دود 16](#_Toc529789561)

[MQTT 17](#_Toc529789562)

[معماری 17](#_Toc529789563)

[کیفیت خدمات 18](#_Toc529789564)

[امینت 18](#_Toc529789565)

[شکل 1 شماتیک برد Programmer 7](file:///C:\Users\DearUser\Desktop\IoT\report.docx#_Toc529881499)

[شکل 2 شماتیک بخش اصلی 7](#_Toc529881500)

[شکل 3 شماتیک بر تنظیم دما 8](file:///C:\Users\DearUser\Desktop\IoT\report.docx#_Toc529881501)

[شکل 4 شماتیک برد روشنایی 8](file:///C:\Users\DearUser\Desktop\IoT\report.docx#_Toc529881502)

[شکل 5 شماتیک برد دما و تشخیص دود 9](file:///C:\Users\DearUser\Desktop\IoT\report.docx#_Toc529881503)

[شکل 6 الگوی publish/subscribe در پروتکل MQTT 10](#_Toc529881504)

# چکیده

با پیشرفت روزافزون تکنولوژی و گسترش حوزه‌ی اینترنت اشیا به منظور ایجاد سهولت در زندگی روزمره، تصمیم گرفتیم با هوشمندسازی محل کار در قسمتی از این پروژه‌ی جهانی سهیم باشیم.

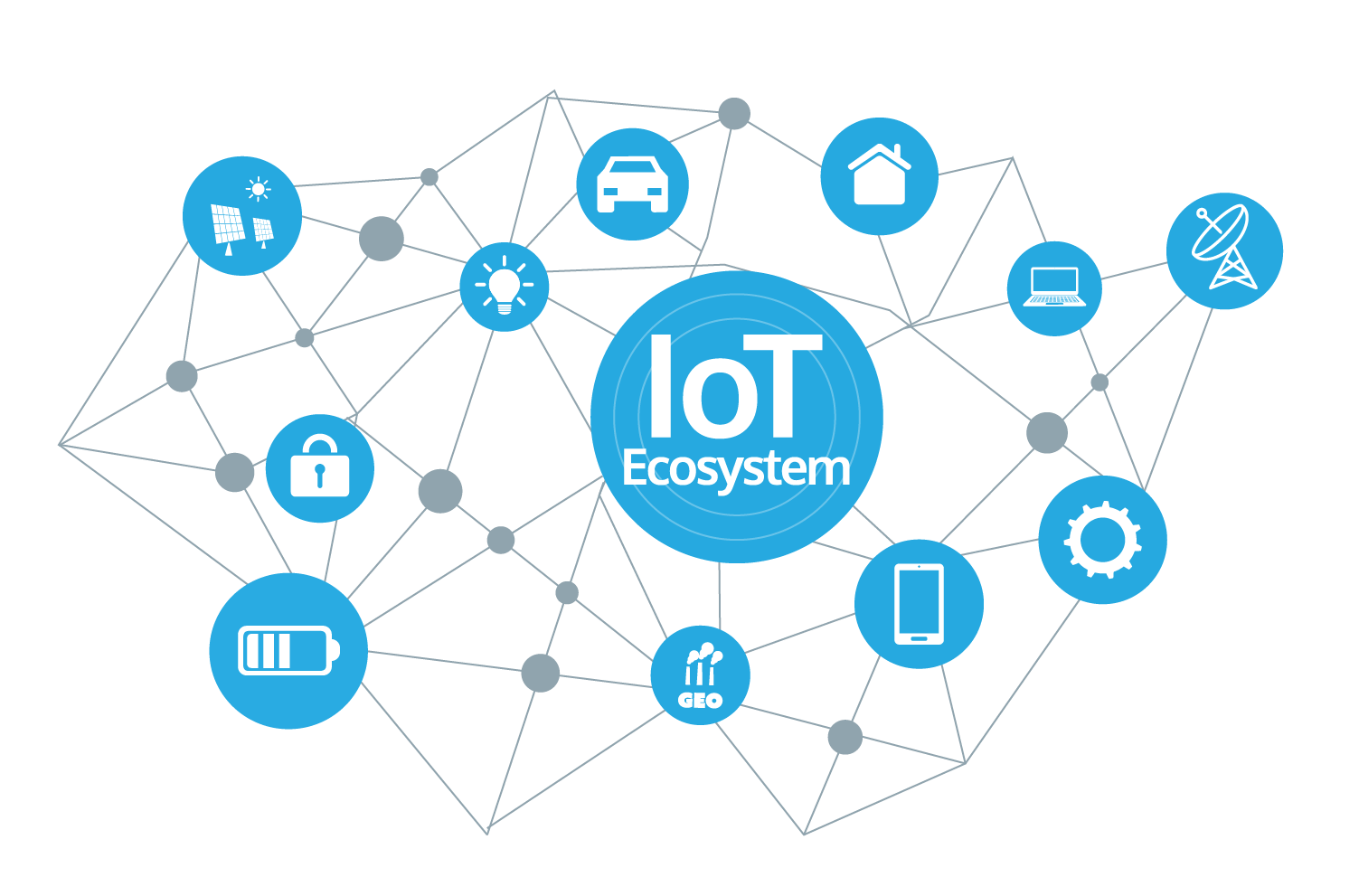
محل کار از بخش‌های مختلفی تشکیل می‌شود که قابلیت هوشمندسازی را دارند. با استفاده از قابلیت‌ها و کمک گرفتن از پروتکل‌های ارتباطی از جمله ام‌کیو‌تی‌تی (MQTT) و دیگر پروتکل‌های ارتباطی از راه دور و هم چنین سنسورهای مختلف موجود در بازار تلاش خواهیم کرد تا به پیاده‌سازی محل کاری هوشمند نزدیک شویم و در نهایت بتوانیم محیطی مدرن و پویاتر برای کار ایجاد کنیم و آسایش را برای افراد به ارمغان بیاوریم.

این پروژه در بستر رزبری‌پای (Raspberry Pi 3 Model B) انجام می‌شود. رزبری‌پای مینی‌کامپیوتری است دارای ۴۰ پایه، متشکل از ۲۸ پایه‌ی GPIO و ۸ پایه‌ی اتصال به زمین و ۴ پایه‌ی Vcc (شامل ۲ پایه 3.3v و ۲ پایه 5v) است. رزبری‌پای ماژول‌های ارتباطی مثل بلوتوث و وایفای، اترنت و ... را به صورت پیشفرض دارا است. همچنین از توزیع مخصوص سیستم عامل لینوکس ررزبری‌پای، یعنی رزبین (Raspbian) استفاده خواهیم کرد.

در این پروژه قصد داریم با استفاده از رزبری‌پای و به کمک رابط‌های کاربری به پیاده‌سازی محل کاری هوشمند بپردازیم.

# مقدمه

اینترنت اشیا یا IoT[[1]](#footnote-1) مجموعه‌ای است از دستگاه‌ها[[2]](#footnote-2)، وسایل نقلیه، لوازم خانگی، سیستم‌های نهان[[3]](#footnote-3)و دیگر ابزارهای ارتباطی که شبکه‌ای بزرگ را تشکیل داده و می‌توانند با هم در ارتباط باشند.



شکل 1 اکوسیستم اینترنت اشیا

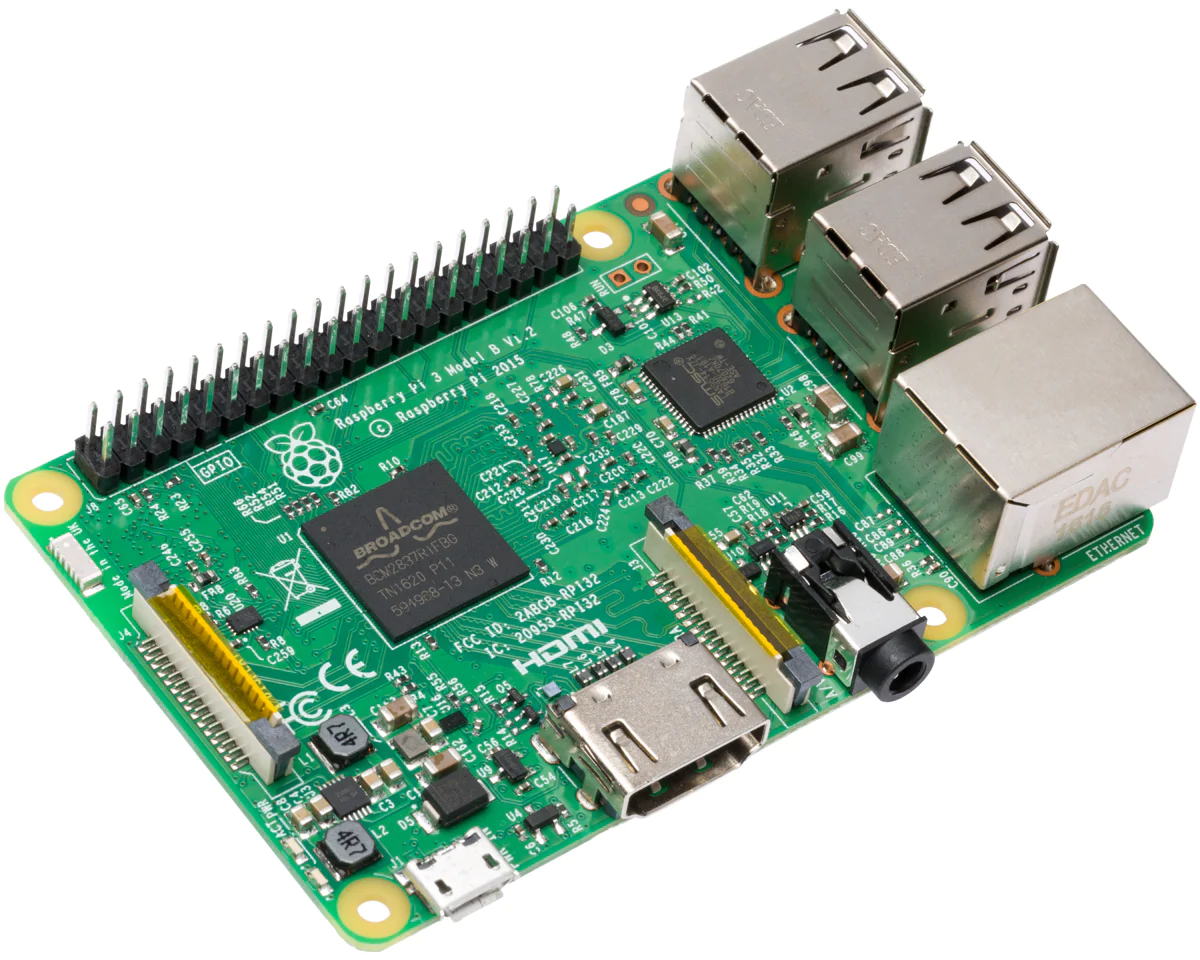
# ریزکنترل‌کننده‌ها

## رزبری‌پای

رزبری‌پای یک رایانه تک-برد در اندازه یک کارت اعتباری است که بنیاد رزبری‌پای آن را ساخته است تا آموزش علوم رایانه را در مدرسه‌ها تشویق کند. سه نسخه از این رایانه ساخته شده که قیمت نسخه‌های اول و دوم به ترتیب ۲۵ و ۳۵ دلار امریکا (به علاوه‌ی مالیات محلی) و نسخه سوم آن با نام zero تنها ۵ دلار است.

این رایانه دارای دیسک سخت نیست ولی به کمک حافظه SD، خود را راه اندازی می‌کند. این رایانه قادر است با سیستم عامل‌های مختلفی راه اندازی شود. سیستم عامل اولیه‌ای که با آن عرضه شد لینوکس نسخه Debian بود اما امروزه نسخه‌هایی از اندروید، ویندوز و ... نیز برای آن تهیه شده است. این رایانه با کمک یک کابل Micro USB تغذیه می‌شود. رزبری‌پای قادر است به کمک یک کابل تصویر ویدئو و یک درگاه سوزنی، خروجی صدا و تصویر داشته و یا با کابل HDMI، خروجی تصویری با دقت ویدیوی HD و صدایی استریو داشته باشد. دو درگاه USB نیز برای آن در نظر گرفته شده است که می‌توان از آن برای ارتباطات مختلف استفاده کرد. از خروجی های USB برای صفحه کلید، ماوس، شبکه بی‌سیم و کلیه دستگاه‌های مورد استفاده از درگاه فوق بهره برده می‌شود. به کمک هاب USB می‌توان از پورت های بیشتری استفاده کرد. نسخه B این رایانه یک خروجی شبکه نیز دارد که به کمک آن می‌توان به شبکه های محلی و یا اینترنت متصل شد. از خروجی HDMI برای صدا، تصویر و کنترل آن با هم می‌توان بهره جست.

در این پروژه از Raspberry Pi 3 Model B استفاده خواهد شد. (شکل ؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟)



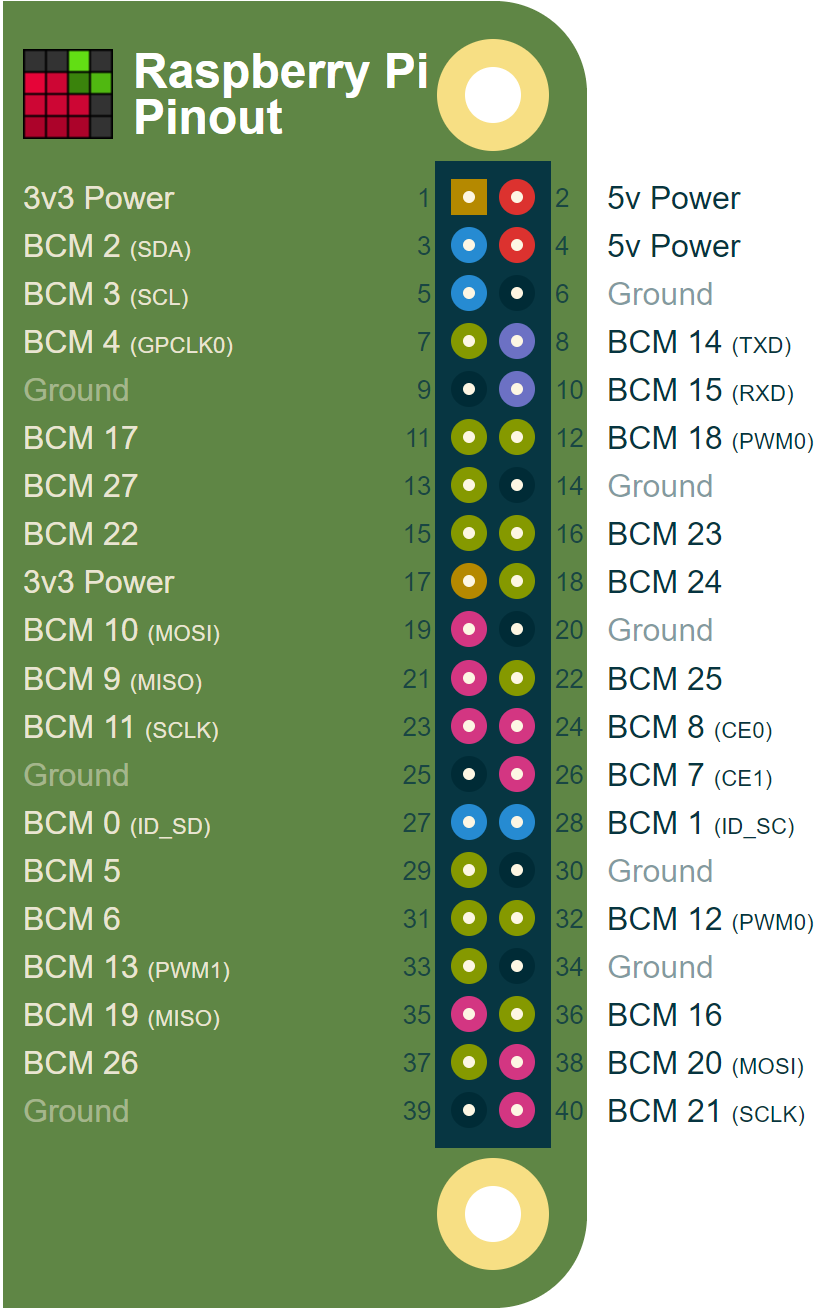
شکل 2 Raspberry Pi 3 Model B

### سخت‌افزار

نسل‌های متفاوتی از رزبری‌پای تا به امروز عرضه شده است که تمامی این مدل‌ها دارای یک SoC[[4]](#footnote-4) شامل دو بخش پردازنده و پردازنده‌ی گرافیکی[[5]](#footnote-5) هستند.

تابه امروز مدل‌های Raspberry Pi Model A/A+، Raspberry Pi 2 و به تازگی Raspberry Pi Zero معرفی شده است. در جدول۱ به جزئیات هر مدل پرداخته شده است.

پین های این مینی کامپیوتر در شکل ؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟ معرفی شده‌اند.



شکل 3 پین‌های خروجی Raspberry Pi 3

جدول 1 مقایسه‌ی مدل‌های متفاوت رزبری‌پای



### سیستم عامل

بردهای رزبری‌پای توانایی اجرای سیستم عامل‌های مختلفی را دارند. در اکثر سیستم عامل‌های لینوکسی برای رزبری‌پای می‌توانید دسکتاپ و دسترسی را تغییر دهید. دانستن این سیستم عامل‌ها به قدرت انتخاب کاربران بر اساس نیاز و استفاده‌شان کمک خواهد کرد.

در این بخش انواع سیستم‌عامل‌هایی که به صورت رسمی توسط بنیاد رزبری‌پای و غیر رسمی معرفی شده اند را بررسی خواهیم کرد.

#### ****Raspbian****

Raspbian سیستم‌عاملی رایگان و بر پایه‌ی لینوکس Debian است. این سیستم عامل توسط بنیاد [Raspberry Pi](https://daneshjookit.com/108-%D8%B1%D8%B2%D8%A8%D8%B1%DB%8C-%D9%BE%D8%A7%DB%8C-raspberry-pi) توسعه داده شده است. سیستم عامل Raspbian بهترین و بیشترین انتخاب کاربران است.  سرعت پاسخ گویی و کارایی سیستم عامل بسیار عالی است. Raspbian از دسکتاپ سبک و کم‌حجم Xfce استفاده می‌کند. حجم این سیستم عامل حدود ۴ گیگابایت است.

در این پروژه از این سیستم‌عامل استفاده خواهیم کرد.

#### ****Google Coder****

این سیستم‌عامل به صورت رسمی از سوی گوگل ارائه شده است. با سیستم‌عامل Google Coder برد رزبری‌پای به یک سرور جهت برنامه‌نویسی HTML، CSS، Java Script و NodeJS تبدیل می‌شود. قابلیت مدیریت فایل‌های هر پروژه و Live Preview نیز در این سیستم‌عامل وجود دارد.

#### ****Android****

سیستم عامل اندروید مخصوص برد رزبری‌پای توسط یکی از توسعه‌دهندگان به صورت غیر رسمی ارائه شده است. آخرین نسخه آن اندروید ۷ Noughat است. حجم این سیستم عامل حدود ۴۰۰ مگابایت است. تقریباً از تمامی قابلیت‌های اندروید می‌توان در برد رزبری‌پای استفاده کرد. ولی سرعت و عملکرد آن نسبت به دیگر سیستم‌های عامل پایین‌تر است.

#### ****Chromium OS****

سیستم‌عامل کرومیوم توسط گوگل رزبری‌پای ارائه شده است. البته به صورت غیررسمی برای رزبری‌پای پورت شده است. حجم آن حدود ۱ گیگابایت است. بهترین کاربرد این سیستم‌عامل سبک، وب‌گردی است.

#### Ubuntu Mate

سیستم‌عامل اوبونتو توزیع‌های مختلفی دارد. Ubuntu Mate نسخه دسکتاپ دارد. این نسخه بسیار سبک است. نسخه Ubuntu Mate مخصوص برد رزبری‌پای دقیقا همانند نسخه دسکتاپ است. Ubuntu Mate بر پایه‌ی Ubuntu بوده و حجم آن ۸ گیگابایت می‌باشد. در مقایسه با سیستم عامل رزبین سرعت آن کمی کُند است. ولی جامعه کاربری تمام تلاشش را برای بهینه سازی انجام داده است.

#### RaspBMC

سیستم عامل گنو/لینوکسی RaspBMC‌، برای تبدیل رزبری‌پای به یک مدیاسنتر خانگی ساخته شده است. کافیست پس از نصب سیستم عامل RaspBMC برد را به تلوزیون وصل کنید. از برخی از قابلیت‌های این سیستم عامل می‌توان به پشتیبانی از زیرنویس فارسی، پشتیبانی از اکثر فرمت‌های ویدئویی، پشتیبانی از فیلم‌های با وضوح ۱۰۸۰p و پشتیبانی از تماشای ویدئو به صورت آنلاین اشاره کرد.

#### Retropie

این سیستم‌عامل بهترین سیستم‌عامل برای تبدیل رزبری‌پای به یک کنسول بازی کامل می‌باشد. این سیستم‌عامل نیز بر پایه‌ی دبیان است و آخرین نسخه از آن حدود ۲ گیگابایت حجم دارد. Retropie به صورت پیش‌فرض با دسته‌های بازی[[6]](#footnote-6) کار می‌کند؛ حتی از دسته‌های Playstation و Xbox نیز پشتیبانی می‌کند. پکیج‌های مختلفی از شبیه‌سازهای بازی بر روی Retropie قابل نصب هستند. بازی‌های قدیمی‌ای مانند Doom، Warcraft، Duke Nuke و … نیز به صورت پورت شده بر روی این سیستم‌عامل قابل اجرا می‌باشند.

Lakka  
این سیستم‌عامل مانند Retropie بوده؛ اما با قابلیت‌ها و حجم کم‌تر. سرعت رابط کاربری آن نیز کمی از Retropie بیش‌تر است. حجم آن حدود ۱۴۰ مگابایت می‌باشد. لاکا فقط از کنسول‌های معروف مانند Play Station 1 و PSP پشتیبانی می‌کند.

#### ****RISC OS Open****

سیستم عامل RISC OS مخصوص پردازنده‌های ARM طراحی شده است. این سیستم عامل برپایه هیچ یک از سیستم عامل‌های لینوکس و یونیکس نمی‌باشد.

#### ****Windows 10 IoT Core****

سیستم عامل IoT Core روی برد رزبری‌پای و Minnowboard Max قابل اجراست. سیستم عامل ویندوز IOT روی دستگاه‌های بدون نمایشگر هم کار کند. از آنجایی که IoT Core نسخه کم حجم شده ویندوز ۱۰ است؛ در آن رابط کاربری پنجره‌ای و دسکتاپ را نمی‌بینید. بلکه توسعه دهندگان باید اپلیکیشن فراگیر را پیاده سازی نمایند که نقش رابط کاربری را عهده دار شود.

#### ****PiNet****

سیستم عامل پای نت با نگرش استفاده در کلاس‌های درس طراحی شده است. یک رزبری‌پای به عنوان سیستم مرکزی با سیستم عامل پای نت PiNet به عنوان سیستم مرکزی و دیگر بردهای رزبری‌پای در کلاس درس به آن متصل می‌شوند. هر دانش آموز دارای نام کاربری و گذر واژه خواهد بود. با این روش فعالیت‌‌های کلیه دانش آموزان قابل ردیابی است. همچنین طراحی سیستم عامل پای نت PiNet کلا از کف انجام شد است. از این رو تمامی نکات مورد نیاز در کلاس درس در آن اعمال شده است.

#### ****Noobs****

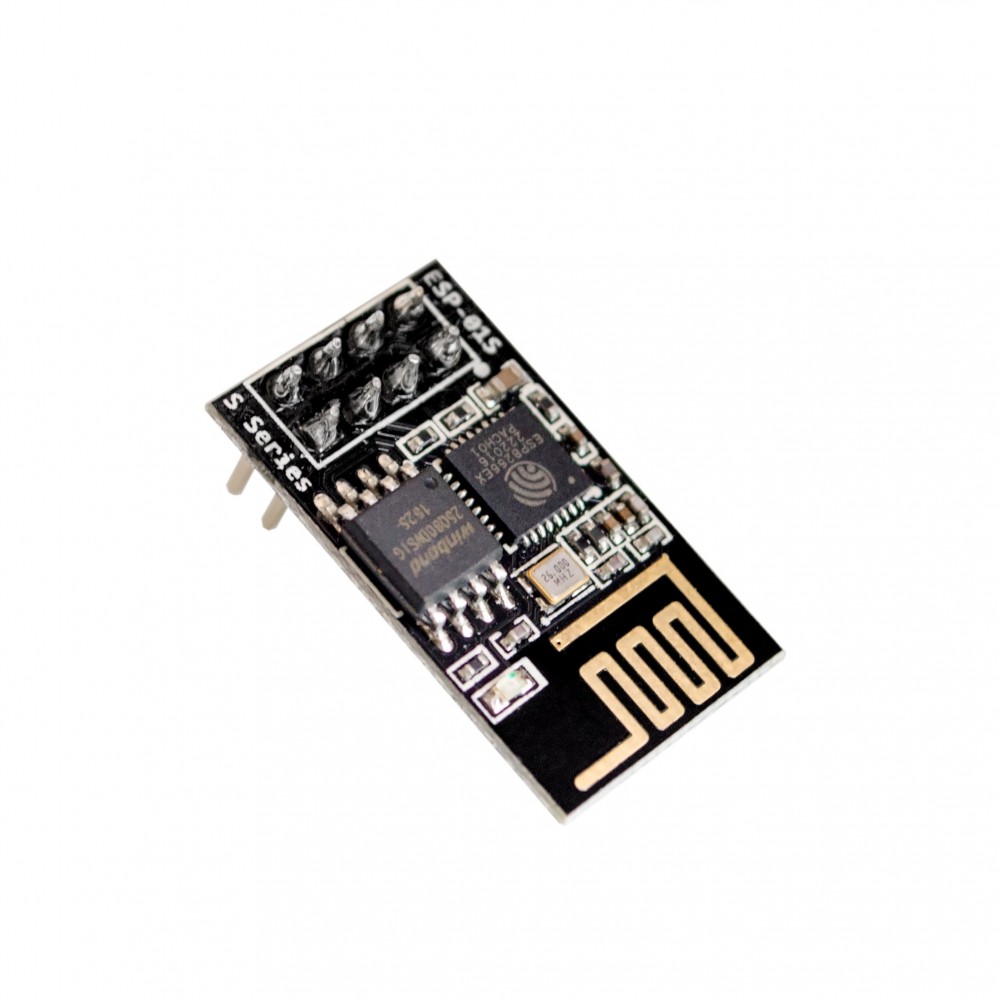
پیشنهاد بنیاد رزبری‌پای برای تازه کارها استفاده از سیستم عامل نوبز Noobs است. این سیستم عامل سبک و سریع است. نوبز Noobs مخفف New Out Of Box Software می‌باشد. نصب نوبز بسیار ساده است و پس از نصب آن می‌توان عامل‌های مورد نظر را از لیست انتخاب کرد تا خودشان اتوماتیک نصب شوند. به عبارتی یک رابط برای نصب سیستم عامل‌های دیگر است.

## ESP8266

ESPها میکروکنترلرهای کوچک و ارزان قیمتی هستند که می‌توانند به شبکه‌ی Wi-Fi وصل شده و به راحتی ارتباطی از جنس TCP/IP برقرار کنند. با استفاده از این ماژول به راحتی می‌توان دستگاه‌های الکترویکی را از راه دور و از طریق اینترنت کنترل کرد. این ماژول‌ها توان مصرفی کمی دارند.

ماژول‌های ESP انواع متفاوتی دارند که تمامی آن‌ها از یک پردازنده‌ی یکسان استفاده می‌کنند و تفاوت اصلیشان در تعداد پایه‌های در دسترس است. هر نوع فواید و مضراتی دارد که بسته به هدف پروژه باید انتخاب شود. در این پروژه از ماژول ESP8266-01 استفاده شده است.

این نوع ESP یکی از پر استفاده‌ترین انواع ESP است که البته کار با آن به سادگی دیگر انواع آن نیست. این نوع ESP ابعاد کوچکی داشته و به نسبت دیگر مدل‌ها ارزان‌تر است. دارای ۸ پایه بوده (شکل ؟؟؟؟؟؟؟؟؟؟؟؟؟؟) و به راحتی در بازار ایران یافت می‌شود.

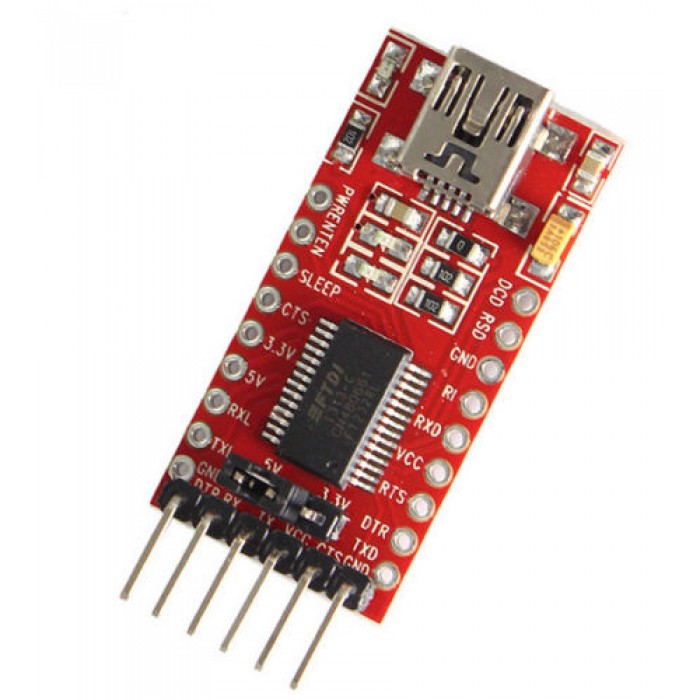


شکل 4 ماژول ESP8266-01

در این پروژه قصد داریم با استفاده از یک مودم و پروتکل MQTT اطلاعات را از رزبری‌پای به شبکه‌ی محلی[[7]](#footnote-7) منتقل کرده و ESP از شبکه دریافت کرده و اطلاعات را از طریق GPIO به دستگاه متصل به آن می‌دهد. (دقت شود که این پروژه مبنی بر شبکه‌ی محلی است و از اینترنت استفاده‌ای نمی‌کند).

### برنامه‌ریزی

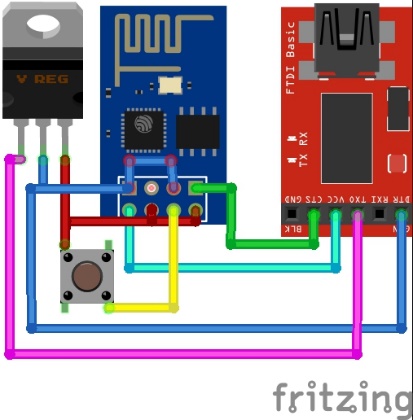
ESPها با استفاده از روش‌های متفاوتی قابل برنامه‌ریزی هستند که راحت‌ترین روش آن استفاده از مبدل USB به TTL (شکل ؟؟؟؟؟؟؟؟؟؟؟؟؟؟) است.



شکل 5 مبدل USB به TTL

مبدل USB به سريال، ابزاری است که برای ارسال و دریافت داده‌ها از یک رابط UART مورد استفاده قرار می‌گیرد. این مبدل به برنامه‌نویسی برای میکروکنترلرها ندارد و قابلیت ارسال و دریافت داده‌ها را بین ماژول های UART مانند GSM, GPS, بلوتوث، Wi-Fi و ... و همچنین كامپيوتر از طریق پورت USB را دارد. با استفاده از مبدل USB به سريال ، به راحتي مي‌توانيد پروژه يا محصول خود را به رابط USB مجهز كنيد. این مبدل، امكان برقراري ارتباط با كامپيوتر از طريق پورت USB را فراهم مي‌كند. در حقيقت با اتصال اين مبدل به كامپيوتر و نصب برنامه درايور، يك پورت سريال مجازي به پورت‌هاي كامپيوتر افزوده مي‌شود. اين مبدل قابليت كار با سيستم عامل‌هاي متفاوت را دارا مي‌باشد.‌ با استفاده از اين ماژول بردهايی كه فاقد پورت USB هستند را به راحتی می‌توان برنامه ريزي كرد.

برای اتصال ESP8266-01 به مبدل کافی است به شیوه‌ی زیر عمل کنید (جدول ؟؟؟؟؟؟؟؟؟؟؟؟؟؟ و شکل ؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟):



شکل 6 شماتیک برد Programmer

جدول 1 نحوه‌ی اتصال ESP8266-01 به مبدل USB-to\_TTL

|  |  |
| --- | --- |
| USB-to-TTL Converter | ESP8266-01 |
| RX | TX |
| GND | GND |
| VCC | CH\_PD |
| - | GPIO2 |
| Button | RST |
| GND | GPIO0 |
| VCC | VCC |
| TX | RX |

هنگامی که ESP در حالی روشن شود که پایه‌ی GPIO0 آن به زمین وصل باشد، ESP به حالت برنامه‌ریزی شدن رفته و آماده‌ی نوشتن روی حافظه‌ی خود می‌شود. پس حتماً دقت شود که قبل از اتصال پایه‌ی VCC به ولتاژ ۳.۳ ولت، پایه‌ی GPIO0 به زمین وصل شده باشد.

دقت شود که بیشتر از ۳.۳ ولت به ESP نباید ولتاژ داده شود. پس ولتاژ مبدل حتماً باید روی ۳.۳ ولت تنظیم شود.

در صورتی که از دکمه[[8]](#footnote-8) استفاده نمی‌کنید، پایه‌ی Reset را مستقیماً به VCC وصل کنید.

### معماری

این میکروکنترلر ۳۲ بیتی بوده و معماری آن RISC[[9]](#footnote-9) است. ESPها توان مصرفی بسیار کمی دارند و بیشترین مقدار کلاک آن‌ها به 160 MHz می‌رسد. همچنین دارای 1 MiB حافظه‌ی Flash، 32KiB حافظه مختص دستورات[[10]](#footnote-10)،

# صفحه نمایش

در این پروژه از یک آینه‌ی هوشمند به عنوان صفحه نمایش‌گر پروژه استفاده خواهیم کرد.

# سنسورها

### کارتخوان

تراشه MF RC522 برای خواندن و نوشتن غیر تماسی تگ‌های RFID در فرکانس 13.56 MHz به کار‌ می‌رود‌. راه اندازی با ولتاژ پایین، قیمت کم‌،انداز کوچک ارتباط بدون تماس چیپ برای خواندن و نوشتن روی کارت و انتقال دو طرفه دیتا با سرعت 424 kb/s این چیپ را برای دستگاه‌های هوشمند و قابل حمل مناسب ساخته است‌. ماژول RC522 دارای قابلیت استفاده آسان با بردهای آردوینو و کاربری مناسب در توسعه ی تجهیزات و کارت خوان‌های RF‌ می‌باشد. این ماژول از یک مودلاسیون برجسته و دمدوله کردن کاملا یک پارچه برای ارتباط بدون تماس در محدوده فرکانسی 13.56 MHz بهره‌ می‌گیرد‌. این ماژول‌ می‌تواند به طور مستقیم در قالب‌های مختلف ریدر به کار رود.

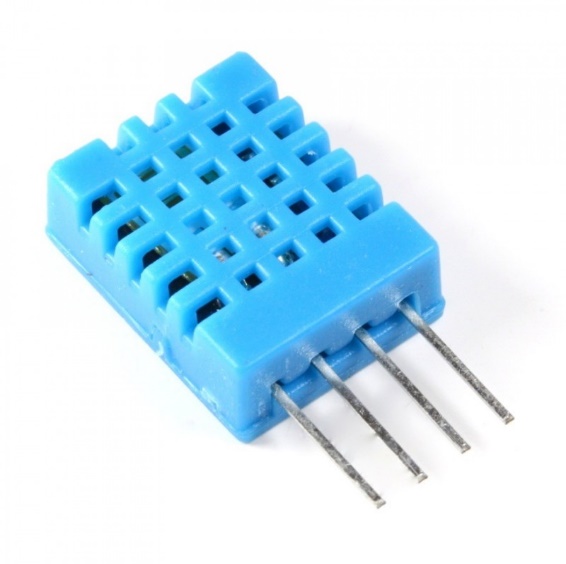
ولتاژ 3.3V، جریان مصرفی 13-26 mA، ارتباط از طریق رابط SPI و سرعت انتقال داده‌ 10 Mb/s از مشخصات این ماژول است. ارتباط برقرار شده توسط این ماژول پایدار و قابل اطمینان است.

### مادون قرمز

### سنجش دما و رطوبت

سنسور رطوبت DHT11 (شکل ؟؟؟؟؟؟؟؟؟؟؟؟؟)، یک سنسور ارزان قیمت جهت سنجش دما و رطوبت هواست که به راحتی می‌تواند به برد آردوینو و یا سایر میکروکنترلرها متصل گردد. اندازه کوچک این سنسور، مصرف کم انرژی و انتقال سیگنال به مسافتی بیش از 20 متر، آن را به یکی از بهترین گزینه‌ها جهت استفاده در پروژه‌های مختلف، بدل کرده است.

البته، نوع دیگری از سنسورهای تشخیص رطوبت و دمای سری DHT به نام DHT22 نیز موجود می‌باشد که نسبت به مدل DHT11 دقیق‌تر و البته گران‌تر است.



شکل 7 سنسور تشخیص دما و رطوبت DHT11

### تشخیص دود

سنسورهای تشخیص دود و گاز سری MQ، نسبت به طیف گسترده‌ای از گازها حساس‌اند و در خانه و دمای اتاق قابل استفاده هستند. این سنسورها از هیتر داخلی کوچک و سنسور الکتروشیمیایی برای تشخیص دود و گاز استفاده می‌کنند. سنسورهای MQ می‌توانند کالیبره شوند اما برای این کار، باید غلظت گاز و یا گازها را بدانند. سنسور گازMQ9 (شکل ؟؟؟؟؟؟؟؟؟؟؟) که در این پروژه از آن استفاده می‌شود، دارای حساسیت بالا نسبت به کربن مونواکسید، متان و LPG است. این سنسور می‌تواند برای تشخیص گازهای شامل CO و گازهای قابل احتراق به کار رود. جنس مواد حساس این سنسور SnO2 است که رسانایی پایینی در هوای پاک دارد. عملکرد تشخیص این سنسور از طریق بالا و پایین رفتن دما است و گاز CO را وقتی که دما پایین می‌آید شناسایی می‌کند. رسانایی سنسور با افزایش غلظت گاز زیاد می‌شود. هنگامی که دما افزایش می‌یابد این سنسور گاز متان و پروپان و دیگر گازهای قابل اشتعال را تشخیص می‌دهد و گازهای دیگر جذب شده در دمای کم را رها می‌کند.

### موتور

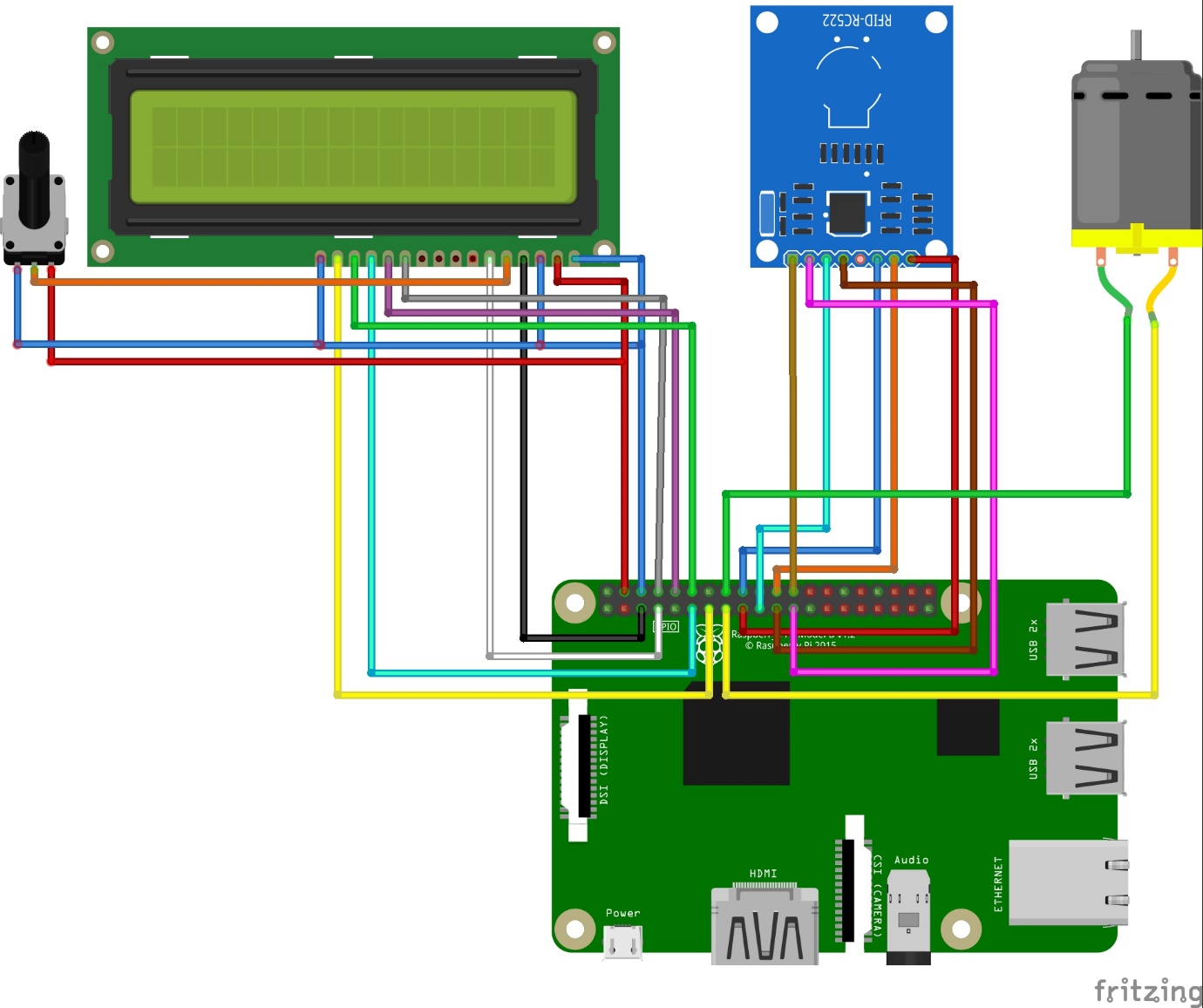
سرو موتور[[11]](#footnote-11) به صورت کلی یک الکتروموتور هست که یک سری مدارات الکترونیکی مانند درایوها در کنار آن قرار دارد و الکتروموتور حاوی شفت عمل گردش را به عهده دارد و تجهیزات الکترونیکی هم وظیفه دقت دادن به الکتروموتور را بر عهده دارند و این دقت شامل کنترل زاویه، کنترل شتاب، کنترل سرعت و ... می‌شود.

سرو موتور ها در انواع گیربکس‌دار و بدون گیربکس وجود دارند و در اندازه های خیلی کوچک تا اندازه‌های بزرگ تولید می‌شوند که اندازه‌های کوچک در پروژه‌های رباتیک و تجهیزات مکاترونیکی استفاده شده و اندازه‌های بزرگ هم در ساخت تجهیزات صنعتی مانند دستگاه‌های CNC استفاده می‌شود.

# بردها

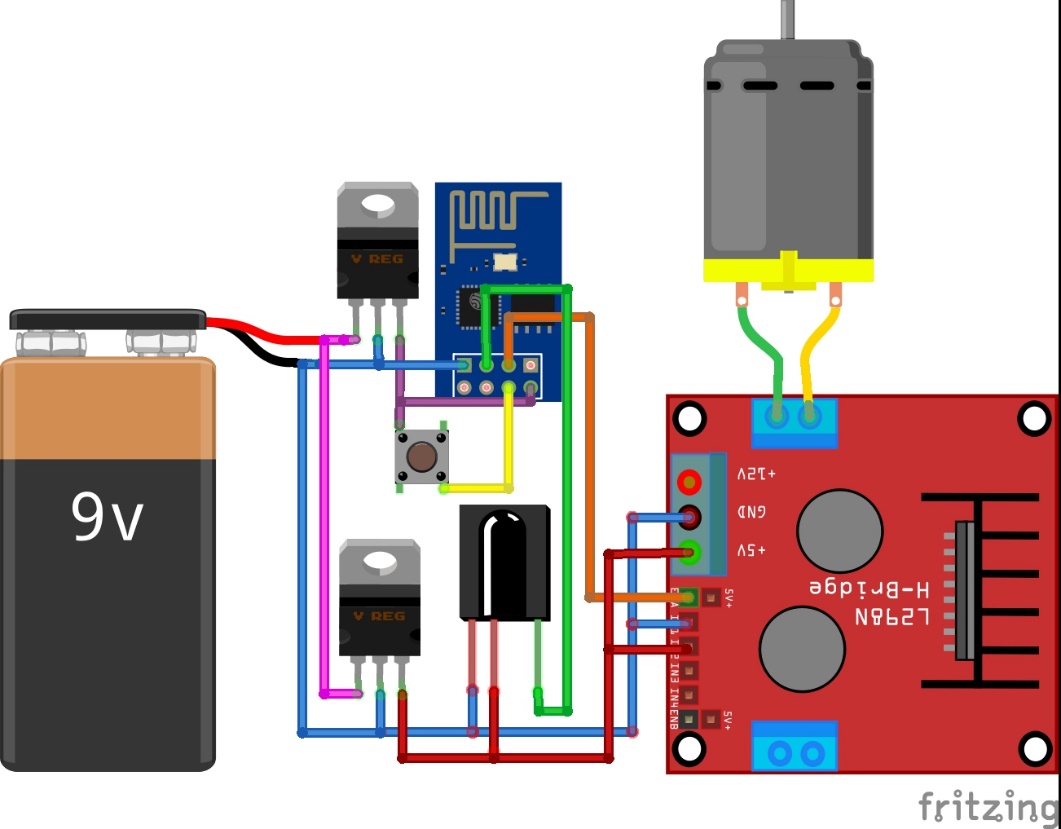
## Programmer

## بدنه‌ی اصلی



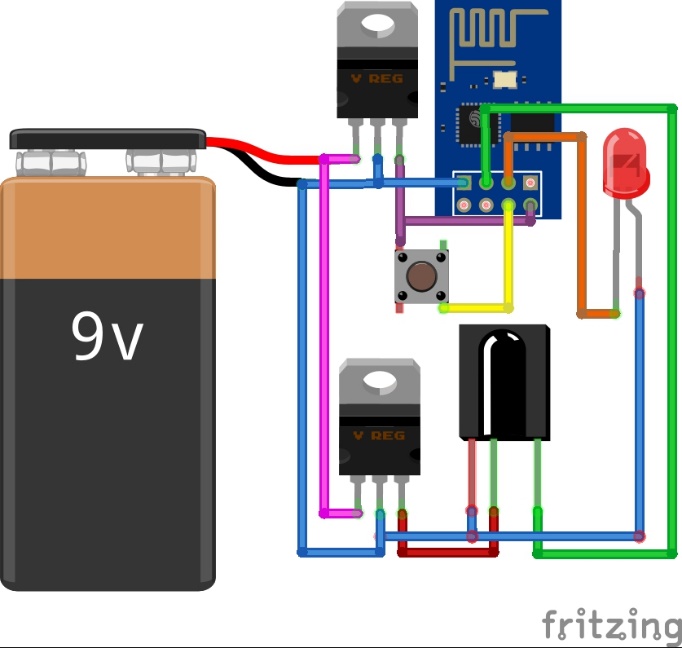
شکل 8 شماتیک بخش اصلی

## تنظیم دما



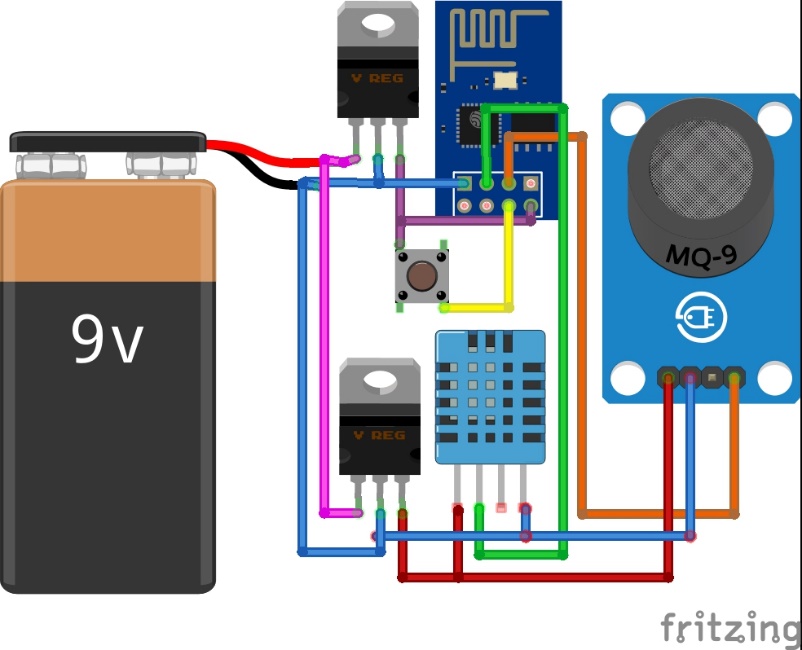
شکل 9 شماتیک برد تنظیم دما

## روشنایی



شکل 10 شماتیک برد روشنایی

## دما و تشخیص دود



شکل 11 شماتیک برد دما و تشخیص دود

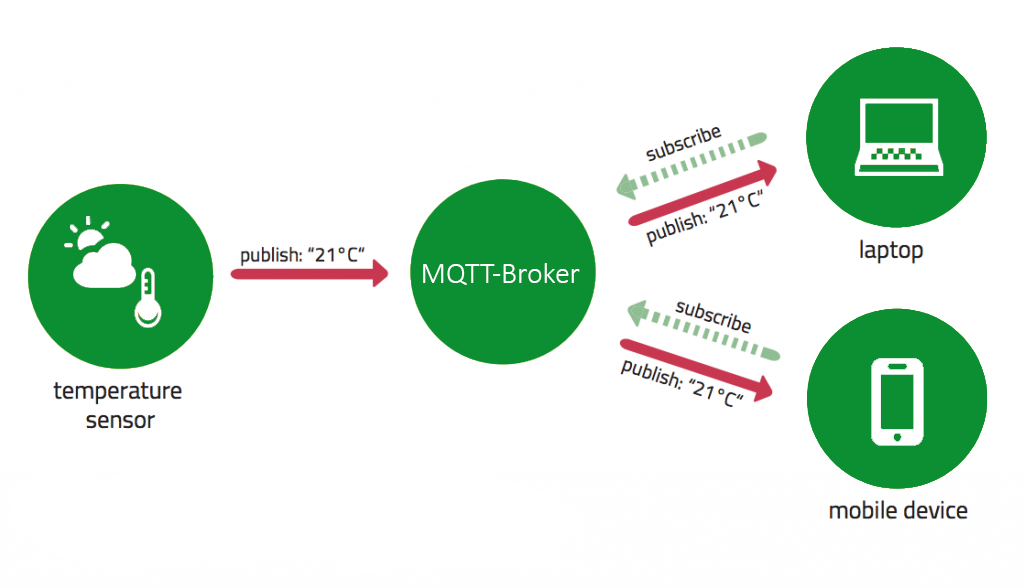
# MQTT

به طور کلی اینترنت اشیا، همانند اینترنت، دارای یک مدل لایه لایه، است. در شکل ۱ می‌توان مدل پیشنهادی اینترنت و اینترنت اشیا را در یک تصویر مشاهده نمود. همانطور که در شکل ۱ مشاهده می‌شود یکی از این لایه‌ها، همانند مدل پیشنهادی اینترنت، لایه کاربرد[[12]](#footnote-12) نام دارد.

پروتکل MQTT[[13]](#footnote-13) یک پروتکل مبتنی بر رویکرد publish/subscribe طراحی گردیده است. این پروتکل در ابتدا به وسیله شرکت IBM توسعه یافت و اکنون یک استاندارد باز می‌باشد. از مزایای این پروتکل به سبک بودن[[14]](#footnote-14)، سازگار بودن با تمام دستگاه‌ها و پیاده سازی ساده آن می‌توان اشاره کرد. این قابلیت‌ها سبب شده است که این پروتکل در موقعیت‌های متفاوتی از جمله ارتباط بین دو ماشین[[15]](#footnote-15) استفاده شود. در ادامه به معرفی بیشتر این پروتکل پرداخته خواهد شد.

## معماری

معماری ساختار MQTT به صورت client/server است. همان طور که در شکل ۲ مشاهده می‌شود، در این پروتکل تعدادی client با توپولوژی ستاره‌ای به یک server که با نام واسط[[16]](#footnote-16) شناخته می‌شود در بستر TCP متصل می‌شوند.



شکل 12 الگوی publish/subscribe در پروتکل MQTT

پروتکل MQTT مبنی بر انتقال پیام است؛ یعنی واسط وظیفه‌ی انتقال پیام‌ها بین client را دارد. هر پیام دارای یک موضوع[[17]](#footnote-17) است. واسط با توجه به موضوع پیام ارسال شده، پیام را به تمامی clientهایی که این موضوع را دارند (به جز client فرستنده‌ی پیام) می‌فرستد. در این پروتکل هر client می‌تواند پیام‌هایی با موضاعات متفاوت را بفرستد و دریافت کند. به عنوان نمونه سنسور دما، دمای محیط را با موضوع “temperature” ارسال می‌کند. واسط پیام را دریافت کرده و به دستگاه یا دستگاه‌هایی که با این موضوع اشتراک[[18]](#footnote-18) می‌فرستد. در صورتی که واسط مشترکی[[19]](#footnote-19) با این موضوع پیدا نکند، پیام را ذخیره کرده و به محض متصل شدن یک کلاینت با این موضوع، تمامی پیام‌های این موضوع را برای client مشترک می‌فرستد. از آن جایی که این پروتکل حافظه‌ی محدودی دارد، تنها قادر ایت پیام‌های اخیر را دخیره کند.

## کیفیت خدمات

پروتکل MQTT، سه سطح برای کیفیت خدمات[[20]](#footnote-20)ارائه می‌دهد. این سطوح عبارتند از:

Fire and Forget: این سطح، تنها پیغام مربوطه را ارسال کرده و خود را درگیر این موضوع که آیا گیرنده پیغام را دریافت کرده‌است یا خیر، نمی‌کند.

Delivered at Least Once: در این سطح، سعی می‌شود تا حداقل یک نمونه از پیغام مربوطه به دست گیرنده برسد.

Delivered Exactly Once: این سطح، همواره یک بسته پیغام را ارسال کرده و منتظر پاسخ ACK آن از سمت گیرنده می‌شود و سپس بسته بعدی را ارسال می‌کند. بنابراین با این سازوکار همواره این اطمینان وجود دارد که از هر بسته دقیقا یک نمونه به دست گیرنده می‌رسد و نیاز به دور ریختن نمونه‌های اضافه‌ی بسته‌ها نیست.

## امینت

برای برقراری ملزومات امنیتی‌، ممکن است واسط از کارخواهان در هنگام برقراری ارتباط، نام کاربری و رمز عبور را درخواست کند. همچنین برای اطمینان از امنیت تبادل اطلاعات در این پروتکل ممکن است ارتباط TCP با پروتکل SSL/TLS رمز شود.

1. Internet of Things [↑](#footnote-ref-1)
2. Devices [↑](#footnote-ref-2)
3. Embedded Systems [↑](#footnote-ref-3)
4. System on a Chip [↑](#footnote-ref-4)
5. Graphic Process Unit (GPU) [↑](#footnote-ref-5)
6. Joystick [↑](#footnote-ref-6)
7. Local Network [↑](#footnote-ref-7)
8. Button [↑](#footnote-ref-8)
9. Reduced Instruction Set Computer [↑](#footnote-ref-9)
10. Instruction RAM [↑](#footnote-ref-10)
11. Servo Motor [↑](#footnote-ref-11)
12. Application Layer [↑](#footnote-ref-12)
13. Message Queuing Telemetry Transport [↑](#footnote-ref-13)
14. Light Weight [↑](#footnote-ref-14)
15. Machine to Machine (M2M) [↑](#footnote-ref-15)
16. Broker [↑](#footnote-ref-16)
17. Topic [↑](#footnote-ref-17)
18. Subscribe [↑](#footnote-ref-18)
19. Subscriber [↑](#footnote-ref-19)
20. Quality of Service (QoS) [↑](#footnote-ref-20)