



پایان نامه دوره کارشناسی مهندسی برق

## طراحی و ساخت سوئیچ ابری از طریق MQTT Server

حسین غلامی

استاد راهنما:

دکتر امیر موسوی نیا

تابستان ۱۳۹۷



لَهُ الْحُكْمُ وَإِلَيْهِ الْمُنْذَرُ



آنچه آموخته ام را تقدیم می‌کنم به

آنان که مهر آسمانی‌شان، آرام بخش دردهای زمینی‌ام است؛

به والدینی که بودنشان تاج افتخاری است بر سرم

که هر چه آموختم در مکتب عشق شما بود و هر چه بکوشم قطره‌ای از

دریایی بیکران مهربانی‌تان را نتوانم سپاس بگویم



ز

## تشکر و قدردانی

از استاد محترم جناب آقای دکتر امیر موسوی نیا که در تکمیل این پروژه از هیچ کمکی دریغ نکردند و زحمت راهنمایی مرا بر عهده گرفتند و همچنین از استاد گرامی ، جناب آقای دکتر یوسف درمانی که زحمت داوری من را متقبل شدند کمال تشکر و قدرانی را می نمایم.



## چکیده

در این پروژه هدف اصلی طراحی یک پریز هوشمند است، که قابلیت کنترل بر بستر اینترنت را داشته باشد، به این معنی که بتوان با هر وسیله‌ای که توانایی وصل شدن به اینترنت را داشته باشد(موبایل، کامپیوتر و...) بتوان پریز هوشمند را کنترل کرد. علاوه بر آن این پریز هوشمند، توانایی محاسبه توان مصرفی را دارد که به کمک این پارامتر می‌توان، میزان مصرف برق قطعات مختلف را، مهندسی کرد. همچنین این پریز به سنسور دما و رطوبت مجهز می‌باشد و می‌توان از راه دور به شرایط محیطی پیرامون دسترسی یافت. اساس کار پروژه به این نحو است که کاربر به سرور پیام می‌دهد و سرور پیام دریافتی را به میکروکنترلر منتقل می‌کند، میکرو کنترلر در پاسخ یا اطلاعات حسگر‌های خود را به سرور منتقل می‌کند یا پریز‌ها را قطع یا وصل می‌کند.

## فهرست مطالب

عنوان	صفحه
فهرست شکل‌ها	۱
فصل ۱ - مقدمه	۱
۱-۱- پیشگفتار	۱
۱-۲- هدف از این انجام این پروژه	۱
۱-۳- روند نما	۳
فصل ۲ - نگاهی بر اینترنت اشیاء	۵
۲-۱- اینترنت اشیاء	۵
۲-۲- معماری اینترنت اشیاء [۱]	۷
۲-۲-۱- بررسی پروژه انجام شده از دیدگاه فوق	۹
۲-۲-۲- بررسی و مقایسه بین روش‌های مختلف ارتباطی [۲]	۱۱
۲-۲-۳- معرفی سیستم‌های مشابه [۳]	۱۴
فصل ۳ - طراحی سیستم‌ها	۱۶
۳-۱- طراحی مفهومی	۱۶
۳-۱-۱- بررسی بیشتر هر یک از بخش‌های طراحی مفهومی	۱۹
۳-۱-۱-۱- میکروکنترلر ESP32: [۴]	۱۹
۳-۱-۱-۲- حسگر توان : [۵]	۲۱
۳-۱-۱-۳- حسگر دما و رطوبت	۲۳
۳-۱-۱-۴- عملگر روشن/خاموش	۲۵
۳-۱-۱-۵- مبدل AC به DC	۲۵
۳-۱-۲- طراحی سخت افزار	۲۶
۳-۱-۲-۱- حسگر‌های استفاده شده در مدار	۲۶
۳-۱-۲-۲- نقشه‌ی شماتیک محصول نهایی	۲۸
۳-۱-۲-۳- طراحی فیبر مدار چاپی	۳۰
۳-۱-۳- طراحی نرم افزار	۳۱
۳-۱-۳-۱- جریان اجرای برنامه [۹]	۳۱
۳-۱-۳-۲- MQTT چیست؟ [۱۰]	۳۳

۳۴.....	ویژگی های کلیدی MQTT	3-3-2-1-
۳۵.....	MQTT مدل	-۲-۲-۳-۳
۳۶.....	کاربران MQTT	-۳-۲-۳-۳
۳۶.....	سرور(MQTT broker)	-۴-۲-۳-۳
۳۶.....	تایپیک ها	3-3-2-5-
۳۶.....	جلسه(session)	3-3-2-6-
۳۷.....	اشتراک(subscription)	-۷-۲-۳-۳
۳۷.....	ساختار پیام ها در پروتکل MQTT	-۸-۲-۳-۳
۳۹.....	اتصال به سرور و تبادل داده ها	-۹-۲-۳-۳
۴۵.....	انتخاب بروکر مناسب	-۳-۳-۳
۴۵.....	طراحی تایپیک ها و بستر انتقال داده	-۴-۳-۳
۴۸.....	برنامه نویسی حسگرها	-۵-۳-۳
۴۸.....	حسگر توان [7]	-۱-۵-۳-۳
۵۰.....	حسگر دما و رطوبت [8]	-۲-۵-۳-۳
۵۲.....	برنامه ساخته شده	-۶-۳-۳
<b>۵۶.....</b>	<b>فصل ۴ - نتیجهگیری و جمع بندی</b>	
۵۶.....	روش کار	-۱-۴
۵۶.....	مثال های تصویری	4-2-
۶۱.....	اطلاعات عملکردی	-۳-۴
<b>۶۴.....</b>	<b>فصل ۵ - پیشنهادات</b>	
۶۴.....	پیشنهاد شماره ۱	-۱-۵
۶۴.....	پیشنهاد شماره ۲	-۲-۵
۶۵.....	پیشنهاد شماره ۳	-۳-۵
۶۵.....	پیشنهاد شماره ۴	-۴-۵
۶۵.....	پیشنهاد شماره ۵	-۵-۵
۶۵.....	پیشنهاد شماره ۶	-۶-۵
۶۵.....	پیشنهاد شماره ۷	-۷-۵
<b>۶۶.....</b>	<b>فصل ۶ - فهرست مراجع</b>	
<b>۶۹.....</b>	<b>پیوست ۱ هرآن چیزی که برای شروع مورد نیاز است</b>	

۷۱	پیوست ۲ جریان ایجاد برنامه
۷۴	پیوست ۳ مقایسه محیط هایی برای توسعه ESP32
۷۸	پیوست ۴ طراحی جعبه

## فهرست شکل‌ها

صفحه	عنوان
۷	شکل ۱-۲ هرم معماری در اینترنت اشیاء.
۱۳	شکل ۲-۲ مقایسه دروازه‌های متفاوت.
۱۳	شکل ۳-۲ نمودار بصری تفاوت میان دروازه‌ها.
۱۴	شکل ۴-۲ بعضی از سیستمهای مشابه.
۱۵	شکل ۵-۲ بعضی از محصولات مشابه.
۱۹	شکل ۱-۳ طرحی از امکانات ESP32.
۲۲	شکل ۲-۳ نحوه عملکرد حسگر هال.
۲۲	شکل ۳-۳ نحوه عملکرد حسگر هال.
۲۶	شکل ۴-۳ جزئیات بیشتری از حسگر هال.
۲۶	شکل ۵-۳ ماژول و حسگر هال.
۲۷	شکل ۶-۳ حسگر DHT22.
۲۷	شکل ۷-۳ نحوه اتصال میکروکنترلر و حسگر.
۲۸	شکل ۸-۳ شماتیک محصول نهایی.
۲۹	شکل ۹-۳ بخشی از شماتیک محصول نهایی.
۳۰	شکل ۱۰-۳ بخشی از شمانیک محصول نهایی.
۳۰	شکل ۱۱-۳ بخشی از شماتیک محصول نهایی.
۳۰	شکل ۱۲-۳ فیبر مدار چاپی محصول نهایی.
۳۴	شکل ۱۳-۳ نحوه اتصال کاربران سرور MQTT.
۳۵	شکل ۱۴-۳ نحوه تبادل اطلاعات بین کاربران.
۳۵	شکل ۱۵-۳ روند کلی پروتکل.
۳۶	شکل ۱۶-۳ کاربران.
۳۷	شکل ۱۷-۳ فریم کلی در پروتکل MQTT.
۳۸	شکل ۱۸-۳ نحوه تبادل اطلاعات در پروتکل MQTT.
۳۹	شکل ۱۹-۳ انواع مختلف پیام بین کاربر و سرور.
۴۰	شکل ۲۰-۳ مدل پیام ۱ در پروتکل MQTT.
۴۰	شکل ۲۱-۳ مدل پیام ۲ در پروتکل MQTT.
۴۱	شکل ۲۲-۳ مدل پیام ۸ در پروتکل MQTT.
۴۱	شکل ۲۳-۳ مدل پیام ۳ در پروتکل MQTT.
۴۲	شکل ۲۴-۳ کیفیت سرویس دهی نوع .

۴۲.....	شکل ۲۵-۳ کیفیت سرویس دهی نوع ۱
۴۳.....	شکل ۲۶-۳ کیفیت سرویس دهی نوع ۳
۴۳.....	شکل ۲۷-۳ مدل پیام ۳ در پروتکل MQTT
۴۴.....	شکل ۲۸-۳ مدل پیام ۷,۶,۵,۴ در پروتکل MQTT
۴۸.....	شکل ۲۹-۳ بخشی از برگه اطلاعات حسگر هال
۵۰.....	شکل ۳۰-۳ نحوه درخواست به سنسور DHT22
۵۱.....	شکل ۳۱-۳ مفهوم ارسال صفر در ارتباط با DHT22
۵۲.....	شکل ۳۲-۳ مفهوم ارسال یک در ارتباط با DHT22
۵۲.....	شکل ۳۳-۳ صورت کلی ارتباط با DHT22

## فهرست بلوک دیاگرام‌ها

صفحه	عنوان
۱۶.....	بلوک دیاگرام ۱-۳
۱۷.....	بلوک دیاگرام ۲-۳
۱۸.....	بلوک دیاگرام ۳-۳
۱۸.....	بلوک دیاگرام ۴-۳
۴۷.....	بلوک دیاگرام ۵-۳

## فهرست فوچارت‌ها

صفحه	عنوان
۵۳.....	فوچارت ۱-۳
۵۳.....	فوچارت ۲-۳
۵۴.....	فوچارت ۳-۳
۵۵.....	فوچارت ۴-۳

## فهرست جدول‌ها

صفحه	عنوان
۶۲.....	جدول ۱-۴ مشخصات فیزیکی
۶۲.....	جدول ۲-۴ مشخصات ابری
۶۳.....	جدول ۳-۴ مشخصات فنی

س



## فصل ۱ - مقدمه

### ۱-۱- پیشگفتار

چگونه هزینه های محیط خانه و کار خود را به حداقل برسانیم در حالیکه کاهش استفاده از وسایل موردنیاز خود را احساس نکنیم؟ چگونه در عصری که منابع برگشت ناپذیر آبی، فسیلی و ... در وضعیت بحرانی قرار دارند مصرف انرژی را طوری تنظیم کنیم تا به صورت بهینه تمامی نیازهای خود را برطرف سازیم؟

### ۲-۱- هدف از این انجام این پروژه

هدف اساسی از انجام چنین پروژه هایی هوشمند سازی محیط زندگی افراد برای ساده تر شدن زندگی ، افزایش سطح رفاه ، و مهم تر از همه کاهش هزینه ها و کاهش مصرف انرژی است.

تنظیم دمای مطلوب خانه با توجه به نوع استفاده ای که از آن می شود و همچنین امکان مشاهده ولتاژ مصرفی و نوسانات جریان برق به همراه محاسبه و کنترل مصرف انرژی هر یک از لوازم برقی، به سادگی ۳۰ درصد کاهش مصرف انرژی و ۵۰ درصد کاهش هزینه را به دنبال خواهد داشت. اتصال دائمی برخی وسایل برقی، مصرف انرژی و تبعا هزینه ها را افزایش می دهد. شما می توانید پیش از خواب تنظیم کنید کدام یک از وسائل برقی و کدام لامپ ها خاموش شوند تا با روشن شدن هوا راس ساعتی که شما می خواهید اتصال دوباره آن ها برقرار شود. با این کار اولین قدم را در بهینه سازی مصرف انرژی برداشته اید، زیرا بیش از ۲۵ درصد از مبالغ قبوض ماهیانه صرف سیستم روشنایی می شود. هنگامی که به محل کارتان می رسید یادتان می افتد تلویزیون را خاموش کرده اید اما میدانید که دستگاه های الکترونیکی حتی در

حال آمده به کارنیز انرژی مصرف می کنند . بنابراین با گوشی همراه خود جریان برق دستگاه را قطع می کنید تا مجبور نباشد برای انرژی صرف شده ای دستگاه در حالتی که هیچ استفاده ای از آن ندارید، هزینه ای بپردازید. همچنین پس از یک ساعت جریان برق جاروی شارژیتان را قطع می کنید تا پس از تکمیل شارژ دستگاه ،هم طول عمر باتری دستگاه حفظ شده باشد و هم میزان مصرف جریان ورودی برق را کاهش داده باشد. مثال های یاد شده همگی کاربردی از اینترنت اشیاء در زندگی روزمره هستند.

در این پروژه هدف اصلی طراحی یک پریز هوشمند است، که قابلیت کنترل بر بستر اینترنت را داشته باشد. به این معنی که به توان با هر وسیله ای که توانایی وصل شدن به اینترنت را داشته باشد(موبایل، کامپیوتر و...) به توان پریز هوشمند را کنترل کرد. علاوه بر آن این پریز هوشمند ، توانایی محاسبه توان مصرفی را دارد که به کمک این پارامتر می توان ، میزان مصرف برق قطعات مختلف را ، مهندسی کرد. همچنین از مجهز بودن این پریز به سنسور دما و رطوبت می توان یاد کرد که می توان از راه دور به شرایط محیطی پیرامون این پریز دسترسی یافت. که در وسیله های مشابه خود امری جدید است.

این پروژه در دو مرحله از طراحی به انجام رسید ۱- تولید محصول اولیه ۲- تولید محصول نهایی در مرحله اول ، به طراحی بر روی برد های مسی سوراخ دار انجام شد و با اطمینان از نحوه عملکرد ، در مرحله دوم به طراحی فیبر مدار چاپی و تولید محصول نهایی انجامید.

## ۱-۳- روند نما

در این بخش به توضیح روند مراحل نوشتمن پایاننامه پرداخته میشود و محتویات هر فصل بررسی میشود.

در فصل دوم ابتدا به مباحث <sup>۱</sup>IOT و مراحلی که در هر پروژه اینترنت اشیاء میباشد بررسی شود، پرداخته میشود و جایگاه پروژه فعلی در این مبحث انجام میشود. در ادامه به معرفی سایر سیستمها و مقایسه آنها با پروژه انجام شده بررسی میشود. در فصل سوم ، به طراحی سیستم پرداخته میشود و در سه مرحله طراحی مفهومی ، طراحی سخت افزار و طراحی نرم افزار کلیات مباحث توضیح داده میشود.

در فصل چهارم مباحث جمعبندی شده و نتایج مشاهده میشود ، و در فصل پنجم ، پیشنهاداتی که برای توسعه این سیستم بیان میگردد، و فصل ششم مراجع را در خود جای داده است.

---

<sup>۱</sup> اینترنت اشیاء



## فصل ۲ - نگاهی بر اینترنت اشیاء

### ۱-۲ - اینترنت اشیاء

اینترنت اشیاء، شبکه‌ای از اشیاء فیزیکی یا چیزهای تعبیه شده با قطعات الکترونیکی، نرم افزار، و سنسورها و اتصالات است؛ که این اتصالات می‌توانند توسط تبادل اطلاعات با تولید کننده اطلاعات، اپراتورها و یا دیگر دستگاهها ارزش و خدمات تولید کنند. هر شئ به تنها یک توسعه سیستم تعبیه شده، قادر به شناسایی است و همچنین می‌تواند با زیر ساخت اینترنت موجود نیز تعامل داشته باشد. [1]

اینترنت اشیاء مفهومی محاسباتی است، برای توصیف آینده‌ای که در آن اشیای فیزیکی یکی پس از دیگری به اینترنت وصل می‌شوند و با اشیای دیگر در ارتباط قرار می‌گیرند. مفهوم اینترنت اشیاء، به نظر شبیه داستان‌های علمی تخیلی است که در آن یخچال‌ها حرف می‌زنند و خودروها به طور خودکار استارت می‌خورند. اما ارتباط با بستر اینترنت در وسایلی که دائم به یکدیگر متصل هستند، بسیار بیشتر از یک خانه یا خودرو هوشمند بر زندگی ما اثر خواهد گذاشت.

در این تکنولوژی به هر چیز یک شناسه منحصر به فرد تعلق می‌گیرد که بتواند داده‌ها را برای پایگاه داده مشخص شده ارسال کند. دنیای امروزی دنیایی است که در آن، داده و اطلاعات حرف اول را می‌زنند و به نوعی می‌توان به آن‌ها مفهوم طلای مجازی را اطلاق کرد، اینترنت اشیاء نیز با دانستن این مهم، بر پایه داده شکل گرفته و آن را به تمامی اشیای پیرامون محیط زندگیمان بسط داده است. به طور مثال فرض کنید که یخچال منزل شما دارای این تکنولوژی می‌باشد (یعنی امکاناتی تعبیه شده دارد که می‌تواند داده را ذخیره و توسط زیرساخت اینترنت، آن را به پایگاه داده‌ای بفرستد که شما قادر به دیدن اطلاعات آن

باشید) حال فردی از اعضای خانواده‌تان بطری شیر را برداشته و آن را مصرف می‌کند. یخچال نیز بعد از گذشت زمان تعیین شده‌ای، به تمام شدن شیر پی برده و آن را به شما اطلاع می‌دهد.

طبق تعریف ارائه شده در ویکی پدیا عبارت اینترنت اشیا، برای نخستین بار در سال ۱۹۹۹ توسط کوین اشتون<sup>۱</sup> مورد استفاده قرار گرفت و جهانی را توصیف کرد که در آن هر چیزی، از جمله اشیا بی‌جان، برای خود هویت دیجیتال داشته باشد و به کامپیوترها اجازه دهنده، آن‌ها را سازماندهی و مدیریت کنند. اینترنت در حال حاضر همه مردم را به هم متصل می‌کند ولی با اینترنت اشیاء تمام اشیاء به هم متصل می‌شوند. اینترنت اشیاء مفهومی جدید در دنیای فناوری و ارتباطات بوده و به طور خلاصه فناوری مدرنی است که در آن برای هر موجودی (انسان، حیوان و یا اشیاء) قابلیت ارسال داده از طریق شبکه‌های ارتباطی، اعم از اینترنت یا اینترانت، فراهم می‌شود. بستر اینترنت اشیاء بر امواج رادیویی بی‌سیمی قرارداده شده که به دستگاه‌های مختلف این امکان را می‌دهند تا از طریق اینترنت با یکدیگر به برقراری ارتباط بپردازنند. این بستر شامل استانداردهایی مانند وای‌فای، بلوتوث کم‌صرف ZigBee و SIGFOX و LORAWLAN ...<sup>۲</sup> است.

---

<sup>۱</sup> Kevin Ashton

<sup>۲</sup> همگی از gateway های پر کاربرد ، در اینترنت اشیا میباشند، که جلوتر به بررسی تفاوت میان آنها پرداخته میشود.

## ۲-۲ معماری اینترنت اشیاء [1]

معماری در اینترنت اشیاء از یک هرم تبعیت میکند ، که از طریق شکل زیر به توضیح پیرامون هر بخش آن پرداخته می شود.



شکل ۱-۲ هرم معماری در اینترنت اشیاء

### مدل کسب و کار

قبل از اینکه بخواهیم پروژه ای در زمینه اینترنت اشیاء انجام دهیم ، باید مدل کسب و کار و کاربرد اصلی را پیش بینی و طراحی کنیم. در ابتدا می بایست به این سوال پاسخ دهیم که هدف اصلی پردازش ما چیست و از انجام این پروژه سعی در رفع کدام نیاز داریم. در ادامه می بایست اطلاعاتی که قصد پردازش آن را داریم را تجمعیع کنیم ، آنگاه اطلاعاتی را که برای تصمیم گیری نیاز داریم را در شاخه بندی متناسب با تصمیم قرار دهیم تا به بهترین تصمیم برسیم و در ادامه ویژگی های قابل بهبود را بررسی کنیم و برای آن ها راهکار ارائه دهیم.

## سکو یا پلتفرم

میان افزاری است که قابلیت های زیر را داشته باشد :

- امکان ارتباط کاربر نهایی با اشیا هوشمند
- مدیریت داده
- پشتیبانی از برنامه های توسعه دهنده گان
- پردازش و ذخیره متمرکز در ابر

## دستگاه های ارتباطی

دستگاه های ارتباطی به طور کلی چند وظیقه کلی دارند ، که به شرح زیر است:

- ایجاد درگاه امن ، تبادل اطلاعات با پلتفرم
- قابلیت پیاده سازی پروتکل های تبادل اطلاعات با پلتفرم (HTTP,MQTT,REST,...)
- پشتیبانی از پروتکل های تبادل اطلاعات با قطعات الکترونیکی (GPIO,UART,SPI,I2C,CAN)
- ذخیره سازی داده ها به منظور انجام تحلیل های کوتاه مدت

## حسگر ها و عملگر ها

به طور کلی حسگر ها و عملگر ها ، تولید کنندگان اطلاعات و مصرف کنندگان اطلاعات می باشند.

به نحوی که حسگر ها اطلاعات(مثلا دما اتاق) را تولید کرده و از طریق دستگاه های ارتباطی به پلتفرم ارسال می کنند. همچنین دستگاه های ارتباطی متناسب با داده هایی که از پلتفرم دریافت می کنند ، عملگر ها را کنترل می کند و عملگر ها رخ دادهایی در دنیای فیزیکی ایجاد می کنند که منجر به رسیدن ما به هدف مطلوب می شوند.

حسگر ها به طور کلی ابزار ارتباط با دنیای واقعی اند و کمیت های دنیا واقعی را به دستگاه های ارتباطی عرضه می کنند، آن ها می توانند تغییرات فیزیکی و شیمیایی دنیای واقعی را حس کنند، حسگر ها را از طریق توان مصرفی به دو دسته حسگر های توان پایین و توان بالا تقسیم کرد. همچنین حسگر ها را به لحاظ نوع حس کردن به دو دسته حسگرهای همچواری و حسگر های تماسی تقسیم کرد. حسگر ها گاها اطلاعات خام را در اختیار کاربر قرار می دهند (حداقل پردازش) و می بایست داده های آن ها را با دنیای واقعی تطبیق داد (کالیبره کرد).

عملگرها ، ابزار های ارتباط با دنیا واقعی هستند که باعث تغییر در سیستم و محیط می شوند و عملگرها را با توجه به نوع عملکردشان میتوان به صورت زیر دسته بندی کرد:

عملگر های خاموش/روشن ، عملگر های خطی ، عملگر های دورانی با زاویه محدود و عملگر های دورانی پیوسته .

همچنین عملگر ها را به لحاظ نوع شان میتوان به صورت زیر دسته بندی کرد :عملگر های هیدرولیکی ، عملگر های نیوماتیکی و عملگر های الکتریکی.

#### ۱-۲-۲- برسی پروژه انجام شده از دیدگاه فوق

همانطور که پیش تر توضیح داده شد ، مرحله اول هر پروژه اینترنت اشیا ، طراحی مدل کسب و کار است. مرحله کسب و کار

در این مرحله از آماده سازی ، طراحی برنامای یک پریز هوشمند انجام گرفت. هدف اصلی یک پریز هوشمند توانایی قطع و وصل کردن جریان برق می باشد. که برای اطمینان از قطعی و وصلی برق می باشد و با بررسی ویژگی های آن می توان دریافت که با اضافه کردن سنسوری میزان جریان عبوری آن را اندازه گرفت.

با دانستن میزان جریان با فرض ثابت گرفتن ولتاژ ۲۲۰ ولت میتوان توان عبوری لحظه ای را محاسبه کرد.

حال می‌بایست اطلاعات موجود را تجمیع کرد و در شاخه بندی مناسب قرار داد و تصمیم‌های متناسب با آن را گرفت . مثلا: اگر میزان جریان از حدی بیشتر شود به معنی اتصال کوتاه شدن است و کلید باید قطع شود.

### پلتفرم یا سکو

برای پروژه انجام شده در پروپوزال ، پلتفرم گوگل(google iot core) و پلتفرم سامسونگ(Artik) ، منظور شده بود ولی به علت مشکلات سیاسی این پلتفرم ها برای ایرانیان بسته شد. در فصل بعدی در بخش طراحی نرم افزار روند اتخاذ شده به طور کامل شرح داده می‌شود.

### دستگاه های ارتباطی

از ESP32 به عنوان دستگاه ارتباطی استفاده شد ، و از گیت وی ، وای فای برای اتصال به اینترنت بهره برده شد. همچنین از پروتکل MQTT برای اتصال به پلت فرم و انتقال اطلاعات استفاده شد. و برای اتصال سنسور ها هم از پایه های GPIO (برای سنسور هال ) و ADC (برای سنسور هال ) میکرو بجهه‌های کردیم.

### حسگر ها و عملگر ها

با توجه به مدل کسب و کار و ویژگی های بررسی شده ما تنها به سنسور هال برای محاسبه جریان نیاز داریم ولی با توجه به مقاصد آموزشی سنسور دما و رطوبت را نیز به پروژه اضافه شد. پس ما دو حسگر و دو تولید کننده اطلاعات داریم که یکی از آن ها حسگر تماسی(سنسور هال) و دیگری حسگر مجاورتی (سنسور دما و رطوبت) هستند. هر کدام از حسگر ها به طور مجزا در فصل بعدی در طراحی سخت افزار بررسی می‌شوند.

برای عملگر هم، از عملگر روشن/خاموش استفاده شد که از سری عملگر های الکترونیکی است.(مدار رله) که روند طراحی آن هم در فصل سوم بخش طراحی سخت افزار توضیح داده می‌شود.

## - ۳-۲ برسی و مقایسه بین روش های مختلف ارتباطی [2]

در این بخش به معرفی استاندارد های بی سیم اینترنت اشیاء می پردازیم و جنبه های کاری متفاوت آنها را بررسی می کنیم. در ابتدا می بایست چند مسئله و چالش طراحی را برای گره های اینترنت اشیاء بررسی کنیم تا بتوانیم استاندارد ها را بر کاربرد خود ارزیابی کنیم:

۱) هر گره از شبکه اینترنت اشیاء ، توانایی پردازش محدودی دارد و هر پردازشی را نمی توان انجام داد.

۲) سیستم ها بتوانند با باتری کار کنند، تا مسائل تامین انرژی (سیم کشی و ..) به سادگی مرتفع شود و اینکه توان کمی مصرف کنند ، تا نیاز به تعویض باتری نباشد و قابلیت خواب رفتن داشته باشد.

۳) محکم ، و نسبت به مسائل محیطی مقاوم باشند.

۴) تنظیمات ساده ای داشته باشند.

۵) ارزان قیمت باشند، که بتوان تعداد زیادی از آن ها را توسعه داد.

برای بررسی نوع دروازه ارتباطی ، چند راه حل موجود است که به بررسی آن ها می پردازیم:

WiFi (۱)

این دروازه به لحاظ مصرف انرژی ، انرژی زیادی مصرف می کند ولی پهنای باند بالایی دارد

(1-54 Mb/s) و برای کاربردهایی که مشکل تامین توان وجود ندارد استفاده می شود.

BLE<sup>۱</sup> (۲)

این دروازه به لحاظ مصرف انرژی، نسبت به wifi بهتر است ولی بردی مشابه wifi دارد که پهنای باند کمتری را به خود اختصاص میدهد (1 Mb/s).

---

<sup>1</sup> Bluetooth Low Energy

**PAN(Personal Area Networks) (۳)****(802.15.4)ZigBee ▪**

- این دروازه به لحاظ مصرف انرژی، نسبت به wifi و بلوتوث بهتر است ولی بردی بیشتر از هر دوی آن‌ها دارد . ولی پهنانی باند بسیار کمتری(250kb/s) را به خود اختصاص می‌دهد.
- همچنین قیمت آن نسبت به هر دوی آن‌ها مناسب تر است.

**Cellular (۴)****GSM(2G,3G) ,LTE ▪**

- این دروازه به لحاظ مصرف انرژی، نسبت به هم نوعان خود بیشترین مصرف را دارا است
- ولی بردی بیشتر از هر سه‌ی آن‌ها دارد که با پهنانی باند بسیار بالا (بیش از 1.4 Mb/s) بیشترین پهنانی باند را به خود اختصاص می‌دهد. همچنین قیمت مازول‌های آن نسبت به سایرین بیشتر است و اساسا برای پروژه‌های خاصی از IOT که پهنانی باند بالا (مثل انتقال تصویر) نیازمند است، استفاده می‌شود.

**LPWAN(Low Power Wide Area Network)(۵)****LoRaWAN ▪**

- این دروازه به لحاظ مصرف انرژی، نسبت به هم نوعان خود کمترین مصرف را دارا است و بردی بیشتر از همه‌ی آن‌ها دارد ولی پهنانی باند آن بسیار پایین (18 b/s - 37.5 kb/s)
- است و اساسا برای پروژه IOT که نیاز به مصرف انرژی کم دارند، بسیار پر کاربرد است.

**Sigfox ▪**

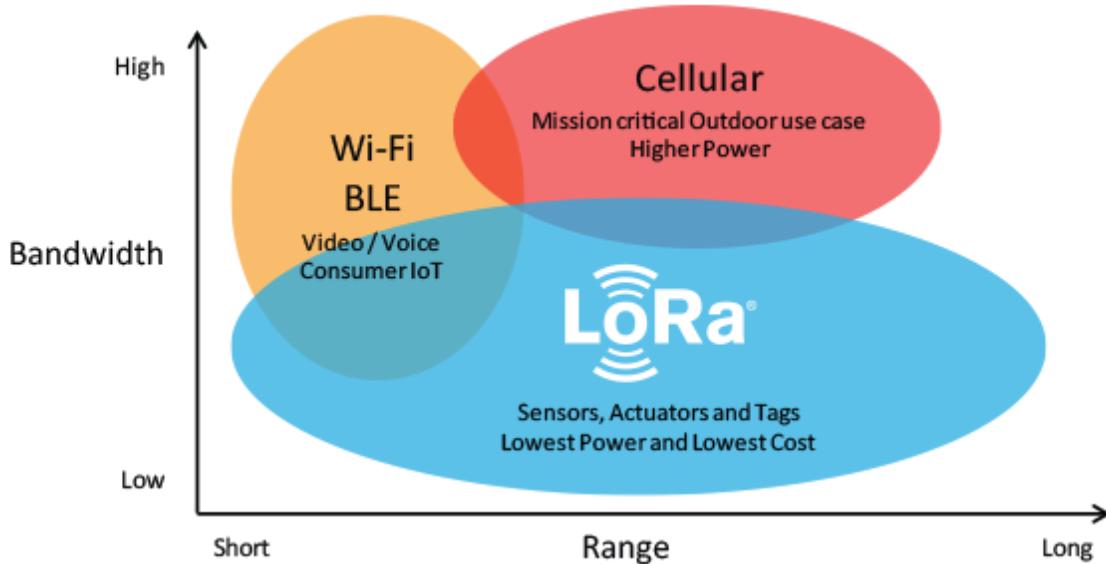
- این دروازه نیز به لحاظ مصرف انرژی، مشابه LoRaWAN و بردی در حدود آن دارد همچنین پهنانی باند آن بسیار پایین (100 b/s) است و اساسا برای پروژه IOT که نیاز به مصرف انرژی کم دارند، بسیار پر کاربرد است، تفاوت آن با LoRaWAN در این است که یک Sigfox شرکت است و آنتن‌ها، پلتفرم و پوشش دهی را خودش تامین می‌کند ولی LoRaWAN به

صورت عمومی عرضه شده و با خرید فرستنده و گره های آن باید شبکه مورد نظر را طراحی کرد.

Technology	Sensitivity	Data rate	Spectrum Strategy
WiFi (802.11 b,g)	-95 dBm	1-54 Mb/s	Wide Band
Bluetooth	-97 dBm	1-2 Mb/s	Wide Band
BLE	-95 dBm	1 Mb/s	Wide Band
ZigBee	-100 dBm	250 kb/s	100 m
SigFox	-126 dBm	100 b/s	Ultra Narrow Band
LoRa	-149 dBm	18 b/s - 37.5 kb/s	Wide Band
Cellular data (2G,3G)	-104 dBm	Up to 1.4 Mb/s	Narrow Band

شکل ۲-۲ مقایسه دروازه های متفاوت

برای جمع بندی ، میتوان نمودار زیر را بررسی کرد، و مقایسه و کاربرد را در این سطح بررسی کرد.



شکل ۳-۲ نمودار بصری تفاوت میان دروازه ها

## - ۴-۲ معرفی سیستم های مشابه [3]

در این بخش به بررسی سایر محصولات مشابه میپردازیم یکی از بزرگ ترین شرکت های دنیا در زمینه اینترنت اشیاء در دنیا ، که در ایران هم فعالیت میکند "شیائومی" است.

در بخش تعدادی از محصولات آن را معرفی می‌شود:

**کتری هوشمند شیائومی (قیمت ۴۰۰ هزار تومان):**

کتری برقی شیائومی بدنه آن به رنگ سفید است و طراحی ساده و ارگونومیک دارد. ظرفیت این کتری ۱,۵ لیتر بوده و قطر سرپوش آن ۱۳ سانتی‌متر بوده و آب را ۵ دقیقه‌ای به جوش می‌آورد. و قدرت گرمایش ۱۸۰۰ واتی آن است. این کتری ضد زنگ و دارای درجه حرارت هوشمند است و با بهینه سازی سیستم برق دستگاه، خطر شوک الکتریکی وجود نخواهد داشت. بوسیله بلوتوث و با استفاده از اپلیکیشن Mi Home میتوان دمای آن را تنظیم کنید.

**جارو برقی هوشمند شیائومی (قیمت ۳ میلیون تومان):**

جارو برقی رباتی شیائومی یک وسیله فوق العاده هوشمند برای تمیز کردن منزل است که به ۱۲ سنسور مجهز می‌باشد. سنسورهای لیزری فاصله، تمامی ۳۶۰ درجه اطراف این وسیله را با سرعت ۱۸۰۰ بار در ثانیه اسکن می‌کنند تا به مانعی برخورد نداشته باشد. نقشه مسیر جارو با هوشمندی طراحی می‌شود. سه پردازنده حرکات دستگاه را به صورت بلادرنگ با الگوریتم محلی کردن و نقشه برداری همزمان (SLAM) محاسبه کرده، بهترین مسیر برای تمیز کردن را ارائه می‌کنند. در مورد شارژ دستگاه نگران نباشید. جارو برقی هوشمند شیائومی دارای یک باتری لیتیوم-یونی ۵۲۰۰ میلی آمپری است که با یک بار شارژ کامل، می‌تواند به مدت ۵/۲ ساعت کار کرده و مساحتی معادل ۲۵۰ متر مربع را جارو کند.



شکل ۴-۲ بعضی از سیستم‌های مشابه

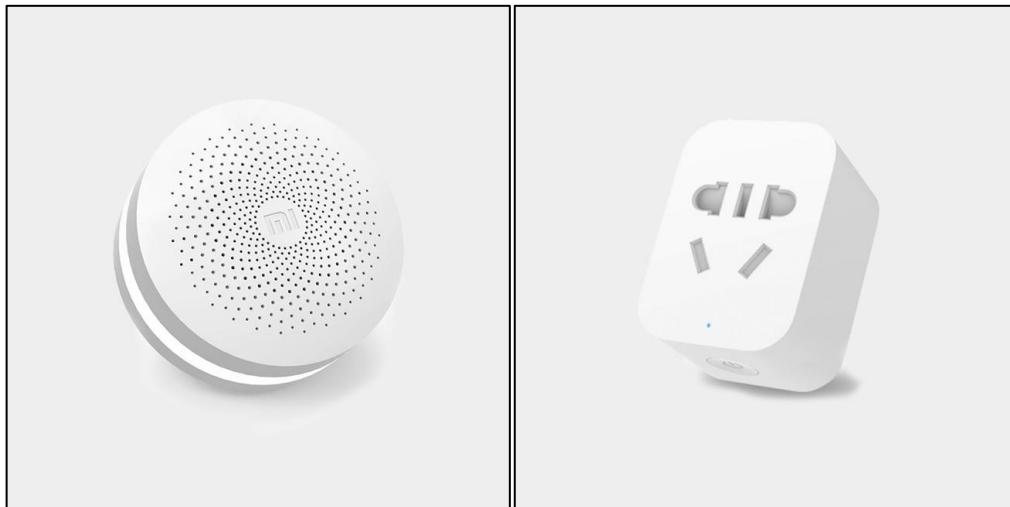
### پریز برق هوشمند شیائومی (قیمت ۱۵۰ هزار تومان):

پریز برق هوشمند شیائومی مدل Smart Socket Plug 2 یکی از محصولاتی است که تجربه لذت استفاده از لوازم خانگی هوشمند را به ارمغان می‌آورد. با این وسیله کوچک و ساده لوازم الکتریکی منزل خود را از هر کجا که هستید کنترل کنید. مهم نیست چقدر از خانه دور هستید، کلید روشن و یا خاموش کردن لوازم منزل در دست شماست. کافی است لوازم برقی منزل خود را به سوکت هوشمند شیائومی متصل کنید. اگر فراموش کرده اید وسیله‌ای را قبل از خروج از منزل خاموش کنید، نگران نباشید. با گوشی موبایل اندرویدی و یا iOS اپلیکیشن خانه هوشمند شیائومی همه جا همراه شماست.

- توجه داشته باشید محصول حاضر مدل زیگ بی است و برای استفاده از آن به سنسور هوشمند

چند منظوره شیائومی مجهر به سیستم هشدار دهنده (سنسور خانه هوشمند شیائومی) نیاز دارید.

**سنسور هوشمند چند منظوره شیائومی مجهر به سیستم هشدار دهنده (قیمت ۲۱۰ هزار تومان):**  
سنسور هوشمند چند منظوره شیائومی مجهر به سیستم هشدار دهنده می‌تواند به عنوان بخش مرکزی برای اتصال سنسورها و دستگاه‌های موجود در مجموعه خانه هوشمند شیائومی عمل کند. با اتصال این دستگاه به سایر دستگاه‌های هوشمند شیائومی می‌توانید مواردی همچون دما، حرکت، نور، صدا و غیره را اندازه‌گیری و کنترل کنید.



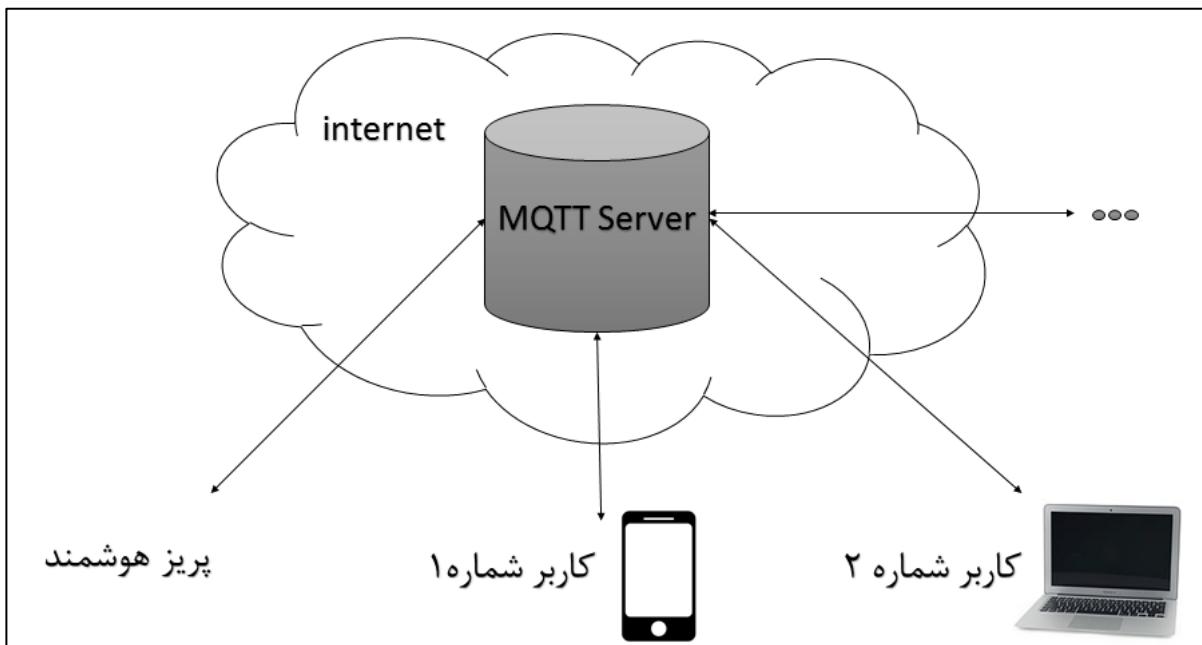
شکل ۵-۲ بعضی از محصولات مشابه

که به این معنا است، در صورتی که بخواهیم پروژه فعلی را با خرید محصولات شیائومی انجام دهیم، به دو پریز هوشمند شیائومی و یک سنسور هوشمند چند منظوره نیازمندیم که ارزش ریالی آن برابر ۵۱۰ هزار تومان می‌شود.

## فصل ۳ - طراحی سیستم ها

### ۱-۳ - طراحی مفهومی

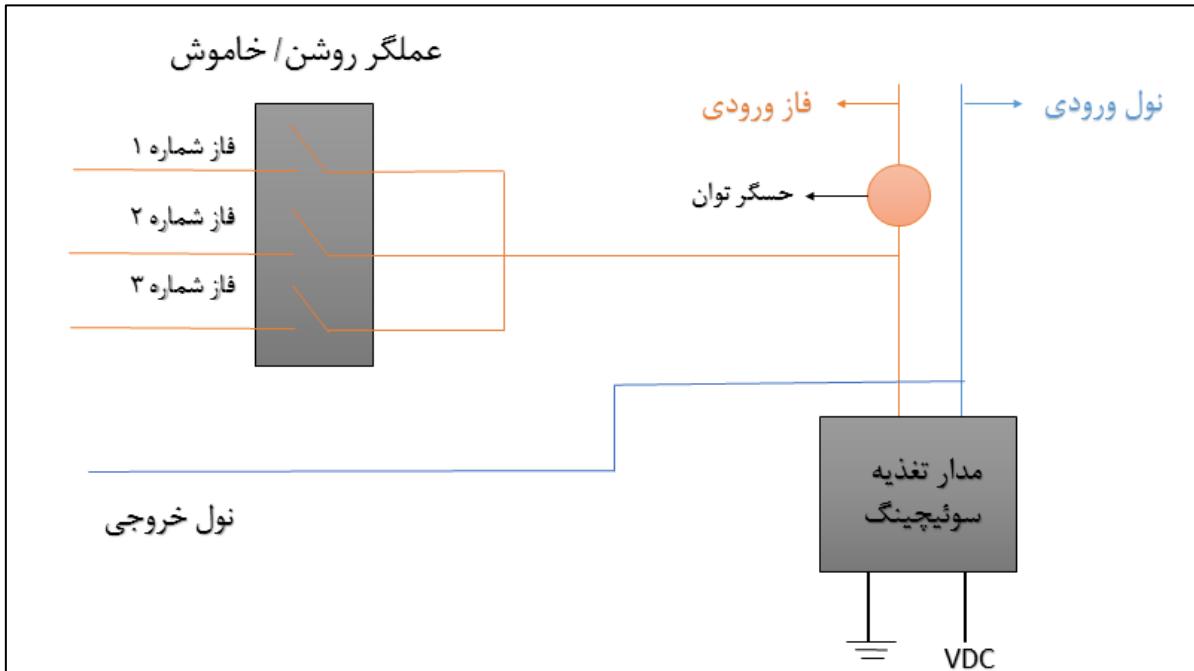
همانطور که در بلوک دیاگرام ۱-۳ مشاهده میکنید ، طرح کلی به این صورت است که پریز هوشمند و سایر کاربران ، به یک سرور MQTT متصل می شوند ، و از این طریق به تبادل اطلاعات میپردازنند.



بلوک دیاگرام ۱-۳ طرح مفهومی

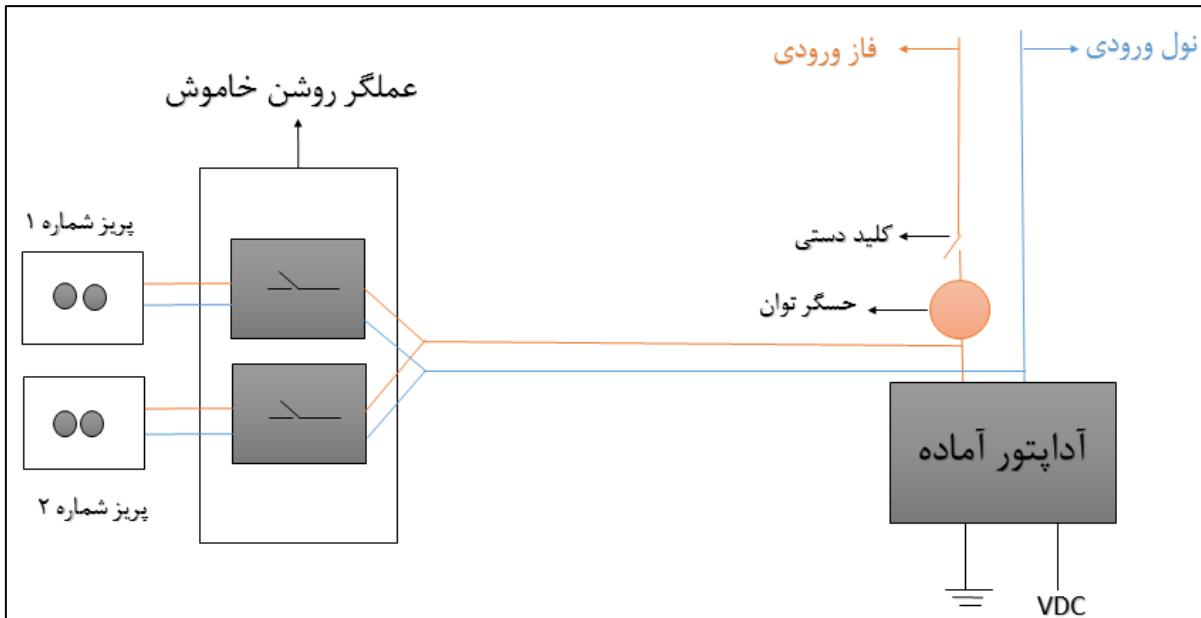
- توضیحات بیشتر پیرامون طرح مفهومی فوق را در بخش طراحی نرم افزار ارائه داده می شود و به همین میزان در این بخش بسنده می کنیم و به بررسی مفهومی سخت افزار پرداخته می شود.
- در بلوک دیاگرام ۲-۳ زیر ، طراحی مفهومی مدار قدرت محصول اولیه را مشاهده می کنیم که نحوه ی فراهم کردن ، انرژی میکروکنترلر و محل قرار گیری حسگر توان و عملگر را نمایش می دهد.

حسگرتوان می‌بایست در ورودی مدار قدرت قرار گیرد تا میزان ، توان کل ، دستگاه (پریز هوشمند) را اندازه‌گیری کند. در طراحی اولیه سیم نول مشترک گرفته شد ، و به تقسیم فاز پرداخته شد. همچنین برای تبدیل برق متناوب به جریان برق یکنواخت از مبدل‌های سوئیچینگ آمده‌ی استفاده شد. MORNSUN



بلوک دیاگرام ۲-۳ مدار قدرت محصول اولیه

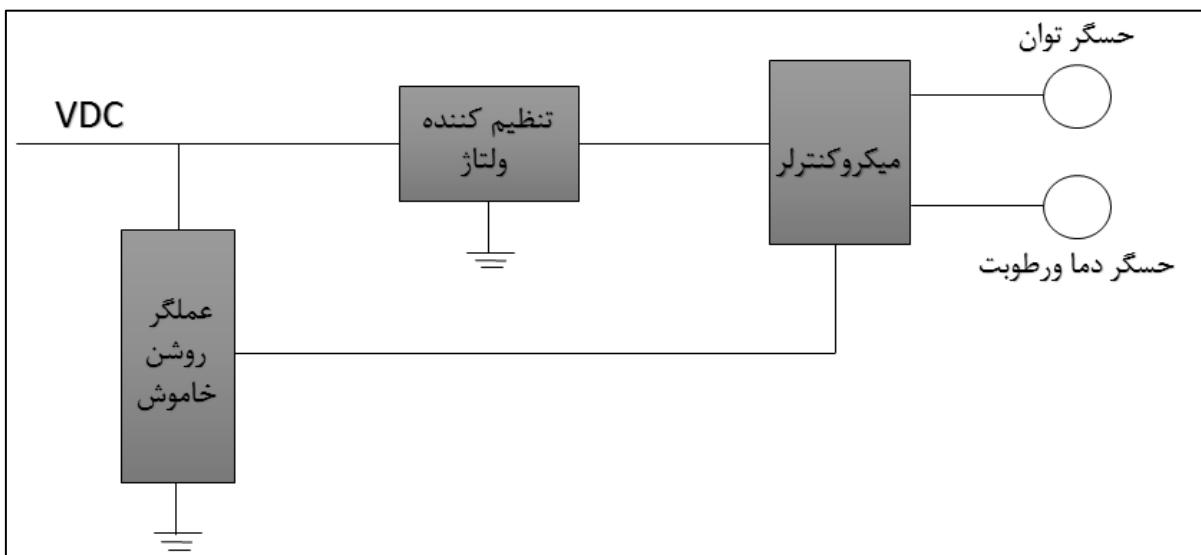
در بلوک دیاگرام ۳-۳ ، طراحی مفهومی مدار قدرت محصول نهایی را مشاهده می‌شود که نحوه‌ی فراهم کردن ، انرژی میکروکنترلر و محل قرار گیری حسگر توان و عملگر را نمایش می‌دهد. در این سری از طراحی یک کلید دستی اضافه شد، تا در خارج از جعبه قرار گیرد ، و همچنین از رله‌های دوتایی ، که فاز و نول را به صورت هم زمان قطع میکنند قرار گرفت. همچنین در محصول نهایی به جای استفاده از مبدل سوئیچینگ به عنوان تبدیل کننده جریان متناوب به جریان یکنواخت ، از یک آداتور آماده استفاده شد و دلیل اصلی آن صرفه‌جویی در هزینه‌ها و کاهش قیمت تمام شده بود.



بلوک دیاگرام ۳-۳ مدار قدرت محصول نهایی

در بلوک دیاگرام ۴-۳ ، مدار کنترل را به صورت طرح مفهومی مشاهده می‌کنیم، که مشخص می‌کند که سیگنالی از حسگر توان که در مدار قدرت قرار دارد، می‌بایست به میکروکنترلر برود و همچنین عملگر روشن/خاموش نیز از میکرو کنترلر فرمان می‌گیرد .

نکته دیگری که می‌بایست در نظر گرفت این است ولتاژ کاری میکرو کنترلر با عملگر متفاوت است و با کمک یک تنظیم کننده ولتاژ<sup>۱</sup> این کار باید انجام شود.



بلوک دیاگرام ۴-۳ مدار کنترل

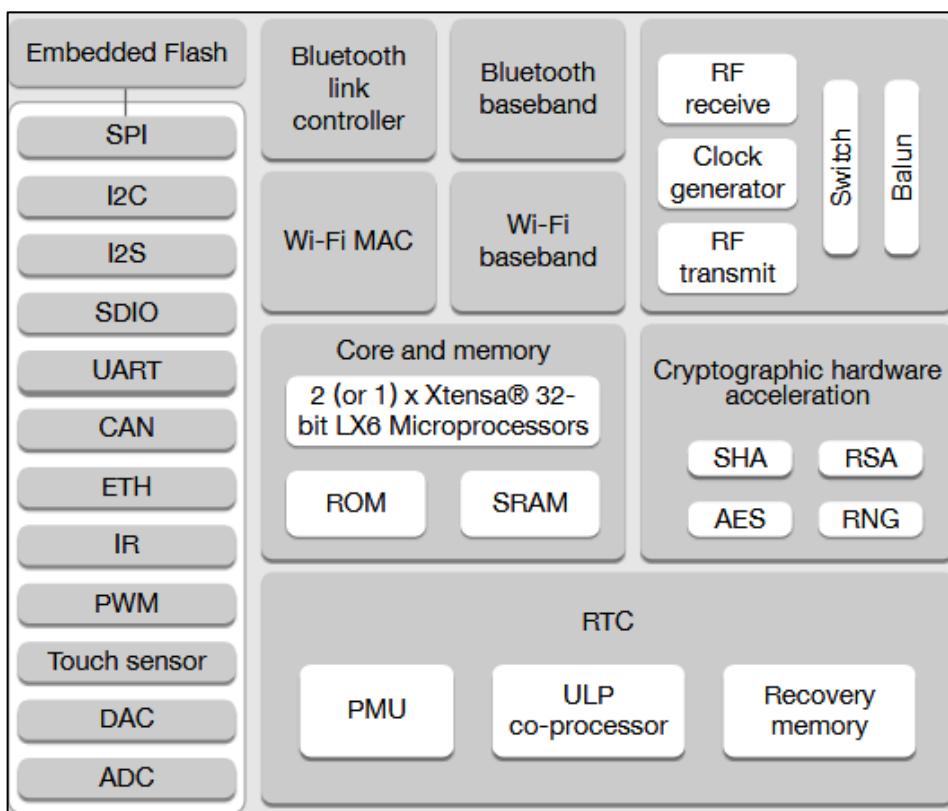
<sup>۱</sup> Voltage Regulator

### ۱-۱-۳- بررسی بیشتر هر یک از بخش های طراحی مفهومی

در این بخش به بررسی جزئیات بیشتر طرح های مفهومی پرداخته می شود و مفاهیمی که در ادامه به آن ها نیاز داریم و طراحی ما را کامل می کند. در این بخش معرفی می شوند.

#### ۱-۱-۱-۳- [4] میکروکنترلر: ESP32

ESP32 از سری میکروکنترلرهای ارزان قیمت و کم توان است. که با WiFi و بلوتوث به تکامل رسیده است. در میکروکنترلر از پردازنده‌ی Tensilica Xtensa LX6 استفاده شده که پردازنده‌ای دو هسته‌ای، یا تک هسته‌ای است که برای مدل‌های مختلف آن متفاوت است. همچنین در آن RF balun نقش آنتن را ایفا می‌کند که با توجه به فرکانس‌های رادیویی ای که ایجاد یا دریافت می‌کند، می‌بایست تقویت کننده توان، تقویت کننده کم نویز برای دریافت سیگنال، فیلتر و مداری برای کنترل توان داشته باشد. که با تکنولوژی ۴۰ نانومتری طراحی شده است.



شکل ۱-۳ طرحی از امکانات ESP32

شکل ۱-۳ نمایی کلی از بخش های موجود در این میکروکنترلر را نمایش می دهد. کاربرد هایی که میتوان برای آن ها میتوان از ESP32 استفاده کرد بسیارند، در اینجا به بخشی از کاربردهایی که در دیتابیت محصول به آن ها اشاره شده پرداخته می شود.

### کاربرد های بسیار کم توان

از این میکرو کنترلر میتوان برای استفاده های قابل حمل ، تکنولوژی های پوشیدنی ، و کاربردهای اینترنت اشیاء بهره برد چرا که با توجه به پردازنده موجود توانایی مدیریت همه بخش های دستگاه را در اختیار داریم مثلا در پروژه انجام شده بلوتوث به طور کامل قطع می باشد و میزان توان فرستاده شده روی آنتن در کمترین میزان خود قرار دارد که باعث می شود این میکروکنترلر در حالت بیداری و ارسال پیام ۱۰۰ میلی آمپر و در حالت خواب تنها ۵۰ میکرو آمپر جریان مصرف کند، که کم مصرف بودن آن را نشان می دهد.

### کاربرد های های کاملاً مجتمع

در طراحی ESP32 از لحاظ توانایی فعال بودن همه بخش ها به طور همزمان خیلی کار شده است. به طوری که به طور هم زمان میکروکنترلر ما می تواند هم از طریق وای فای به یک مودم متصل شود و هم به طور همزمان نقش یک مودم را ایفا کند و در همان زمان بلوتوث خود را روشن و به یک دستگاه دیگر متصل شود ، هم دستگاه ثانویه ای به آن متصل شود. بی آن که تداخلی ایجاد شود. این باعث می شود تا بتوان بین تعداد زیادی میکروکنترلر شبکه ای ایجاد کرد تا همگی با هم تبادل اطلاعات داشته باشند.

دلیل انتخاب این قطعه را می توان توسعه مند بودن آن و امکانات جانبی زیاد آن بیان کرد. چرا که هدف اصلی ، از انجام پروژه ای دانشگاهی افزایش توانایی های فردی است. و با این انتخاب گستره آشنایی ، با انواع مختلف امکانات سخت افزاری و تکنیک های نرم افزاری تجربه فرد را بالا می برد. این در حالی است که با قطعاتی به مراتب ارزان تر میتوانست همین پروژه را با کیفیت فعلی عرضه کرد.

### ۳-۱-۱-۲ - حسگر توان : [5]

حسگر توانی که در بلوك دیاگرام ها وجود داشت، در حقیقت حسگر های جریان هستند. این حسگر ها به علت داشتن عنصر هال (هادی یا نیمه هادی حامل جریان) هنگامی که تحت یک میدان مغناطیسی قرار بگیرند، یک ولتاژ، متناسب با قدرت میدان مغناطیسی تولید می کند. ولی به دلیل کوچک بودن این ولتاژ ، حسگر اثرهال به مداراتی جهت تقویت سیگنال نیز نیاز دارند. سنسور اثرهال یک سنسور میدان مغناطیسی است، ولی کاربردهای دیگری هم دارد. کاربرد های دیگر سنسور هال عبارت اند از : حسگر جریان، حساسیت مجاورتی، حسگر توان، اندازه گیری گرنش، آشکار ساز فلزات، دسته بازی با حساسیت موقعیت میانی، حس کننده درجه حرارت، فشار، خلاء و ...

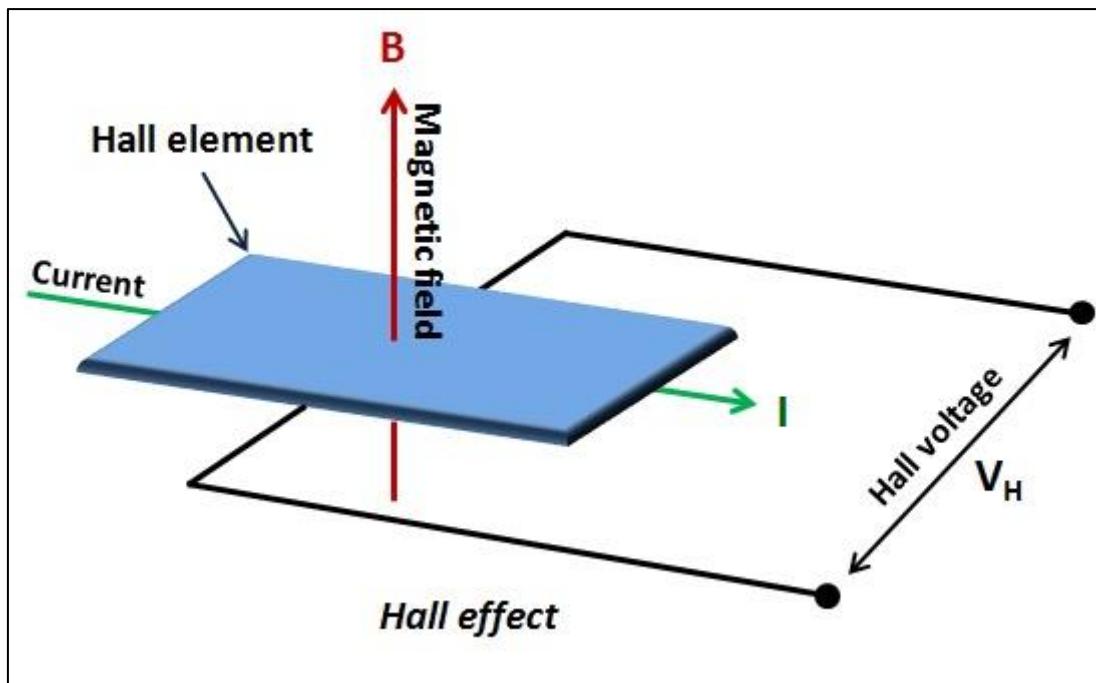
سنسورهای اثرهال به دو نوع عمده تقسیم می شوند:

سنسورهای آنالوگ: ولتاژ خروجی این نوع سنسور مستقیما از خروجی تقویت کننده گرفته می شود که با قدرت میدان مغناطیسی خارجی اعمال شده متناسب است . محدوده های عملکرد این نوع سنسورها می تواند ۴,۵ تا ۴,۵ ولت، ۴,۵ تا ۱۰,۵ ولت، ۱۲ ولت و یا ۶,۶ تا ۱۲,۶ ولت باشد.

سنسورهای دیجیتال: این نوع سنسورها دارای schmitt-trigger هستند که بر اساس یک حلقه هیسترزیس ساخته شده است و به تقویت کننده متصل می شود. خروجی آنها تنها دو وضعیت روشن(ON) و خاموش(OFF) را پوشش می دهد. در صورتی که شار مغناطیسی بزرگتر از یک مقدار مرجع (طبق حلقه ای هیسترزیس) عبور کند خروجی سریعا از حالت خاموش به حالت روشن تغییر وضعیت می دهد. در صورتی که شار مغناطیسی کمتر از مقدار مرجع شود خروجی به حالت OFF می رود.

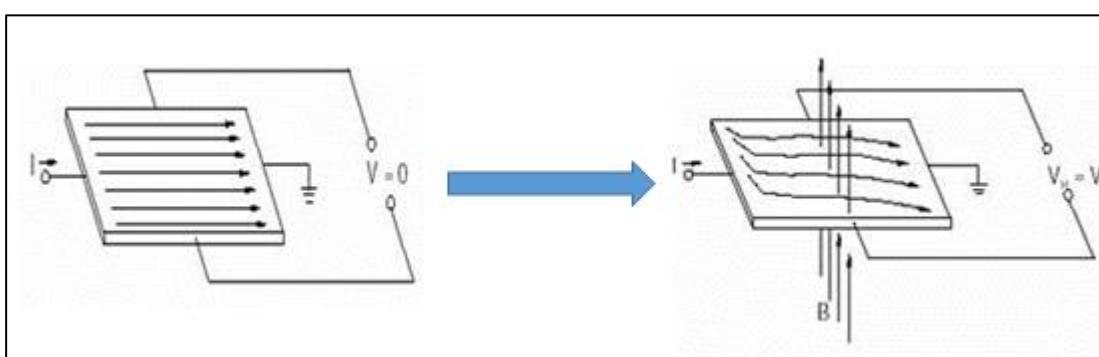
### mekanizm عملکرد:

اگر یک ماده هادی یا نیمه هادی که حامل جریان الکتریکی I است در یک میدان مغناطیسی به شدت B عمود بر جهت جریان عبوری قرار بگیرد، ولتاژی به مقدار  $V$  در عرض هادی القا می شود. علت القا شدن ولتاژ، اعمال نیروی لرنز بر جریان و بر هم زدن توزیع آن است.

$$F = q[E + (V \times B)]$$


شکل ۲-۳ نحوه عملکرد حسگر هال

پس حامل‌های جریان مسیر منحنی را مطابق شکل ۳-۳ می‌بینیم. حامل‌های جریان روی یک لبه قطعه ظاهر می‌شوند، و در لبه مخالف کمبود حامل اتفاق می‌افتد (قانون دست چپ). این عدم تعادل بار باعث ایجاد ولتاژ هال می‌شود.



شکل ۳-۳ نحوه عملکرد حسگر هال

با توجه به هدف ما که محاسبه توان است، با در نظر گرفتن ولتاژ ۲۲۰ ولت و مقدار جریان عبوری میتوان، توان لحظه ای را محاسبه نمود، پس از این ارزیابی به دنبال سنسورهای موجود در بازار ایران پرداخته شد، با اندکی تحقیق و بررسی مژوول های موجود در بازار با ACS712 آشنا میشویم که در مدل های ۵ آمپر، ۲۰ آمپر و ۳۰ آمپر تولید میشود. ابتدا برای محصول اولیه از مژوول ۲۰ آمپر آن و برای محصول نهایی از مدار مجتمع ۲۰ آمپر آن در فیبر مدار چاپی استفاده شد.

### ۱-۱-۳- حسگر دما و رطوبت

#### حسگر دمای مقاومتی (RTD) :

یکی از انواع از حسگر دما، حسگر دمای مقاومتی الکتریکی یا دتکتور دمای مقاومتی یا RTD میباشد. RTD ها حسگرهای دمایی دقیقی هستند که از فلزهای رسانای بسیار خالصی مانند پلاتین، مس یا نیکل به صورت یک پیچه ساخته شده اند، و تابع تغییرات مقاومت الکتریکی آنها بر حسب دما، شبیه به آن چیزی است که در ترمیسترهای وجود دارد. این قطعات دارای یک غشای باریک از خمیر پلاتین هستند که روی یک زیر لایه ای سرامیکی سفید قرار میگیرند. دتکتورهای دمایی مقاومتی دارای ضریب دمایی ثابت (PTC) هستند که خروجی آنها به شدت خطی است که موجب شده به صورت خیلی دقیق دما را اندازه گیری کند. اما آنها دارای حساسیت دمایی بسیار کمی بوده، به طوری که تغییرات دمایی باعث ایجاد تغییرات ولتاژ کوچکی میشود، به عنوان مثال  $\Omega/C1$

نوع بسیار متداول RTD از جنس پلاتین ساخته شده است که دماسنج مقاومتی پلاتینی یا PRT نامیده میشود که مدل پرکاربرد در دسترس از آنها سنسور PT100 است، که دارای مقاومت استاندارد ۱۰۰ اهم در دمای صفر درجه ای سانتی گراد است. درون این قطعات از پلاتین استفاده شده که فلز گران قیمتی میباشد و یکی از معایب این نوع سنسورها قیمت بالای آنها است.

مانند ترمیستور، RTD ها هم قطعات مقاومتی پسیو هستند و با عبور جریان ثابت از درون سنسور دما میتوان یک ولتاژ خروجی به دست آورد که به صورت خطی با دما تغییر میکند. یک RTD معمول دارای

مقاومت مبنای حدود ۱۰۰ اهم در دمای صفر درجه ی سانتی گراد است که در دمای ۱۰۰ درجه به مقدار ۱۴۰ اهم می رسد و محدوده ی دمایی عملکرد آن بین  $-200^{\circ}\text{C}$  تا  $+600^{\circ}\text{C}$  درجه ی سانتی گراد است.

از آنجایی که RTD ها عناصر مقاومتی هستند، نیاز داریم که جریانی از آن عبور دهیم و سپس ولتاژ حاصل را بررسی کنیم. در هر حال تغییرات مقاومت که بر اثر خود گرمایشی قسمت های مقاومتی هنگام عبور جریان ایجاد می شود، IR (قانون اهم) باعث ایجاد خطای شود. برای اجتناب از بروز این مسئله، RTD معمولاً به شبکه ی پل و تستون متصل می شود که دارای اتصالات اضافی برای جبران پیش فاز و یا اتصال به منبع جریان ثابت است.

### حسگر رطوبت خازنی:

سنسور رطوبت ، رطوبت نسبی محیطی که در آن قرار دارد را اندازه گیری می کند. این بدین معنی است که آن هم میزان رطوبت و هم میزان دمای هوا را اندازه گیری می کند. رطوبت نسبی در واقع نسبت رطوبت واقعی موجود در هوا به بالاترین میزان رطوبت هوا در آن دما می تواند داشته باشد است که به صورت درصدی بیان می شود. هر چه هوا گرم تر باشد می تواند رطوبت بیشتری را در خود نگه دارد بنابراین رطوبت نسبی همراه با تغییرات دمای هوا تغییر می کند. یعنی برای تنظیم دقیق سنسور رطوبت به سنسور دما نیز ، نیاز داریم. اکثر سنسورهای رطوبت از روش اندازه گیری میزان ظرفیت خازنی برای تعیین میزان رطوبت موجود در هوا استفاده می کنند. این نوع اندازه گیری بر مبنای دو صفحه که هادی الکتریکی هستند به همراه یک ماده خمیری نارسانا که در بین آنها قرار می گیرد و میدان الکتریکی که بین آنها قرار دارد عمل می کند.

رطوبت هوا در بین این مجموعه قرار می گیرد و باعث تغییر در سطح ولتاژ بین دو صفحه می شود. سپس این تغییر ولتاژ به اندازه گیری دیجیتالی از رطوبت نسبی موجود در هوا همراه با در نظر گرفتن دمای هوا تبدیل می شود.

در بازار ایران سنسورهای مختلفی برای دما وجود دارد، که میتوان به سری‌های LMxx اشاره کرد. همچنین برای سنسور رطوبت میتوان به HR2x که برای بررسی رطوبت خاک(تماسی) است و HS1xx که برای رطوبت هوا(مجاورتی) میباشد ، اشاره نمود.

از بین سنسورهای دما، رطوبت موجود و در دسترس، میتوان به SHT11 و SHT12 که به صورت سطحی هستند و DHT22 و DHT11 که به صورت عمقی هستند یاد کرد.

با توجه به نوع طراحی و طراحی جعبه و اینکه سنسور دما و رطوبت میبایست خارج از جعبه باشد ، سنسور DHT22 که به صورت عمقی بود انتخاب شد تا آن را خارج از جعبه قرار داده و از طریق کانکتور به برد اصلی متصل نمود.

#### ۱-۱-۴-۴- عملگر روشن/خاموش

در روند انجام پروژه دو نوع عملگر خاموش/روشن طراحی شد که به طور کلی ، در محصول اولیه ۳ خروجی ۵ آمپر قرار داده شد و در محصول نهایی دو خروجی ۱۰ آمپر قرار داده شد که شماتیک آن و توضیحات نحوه عملکرد آن و قطعات انتخاب شده در بخش بعد قرار داده شد.

#### ۱-۱-۵-۵- مبدل AC به DC

همانطور که در بلوک دیاگرام ها مشاهده کردید، برای تامین انرژی مورد نیاز میکروکنترلر میبایست ولتاژ ۳,۳ یکنواخت ایجاد کرد. پس در هر دو محصول از نوع خاصی از مبدل ها استفاده شد.

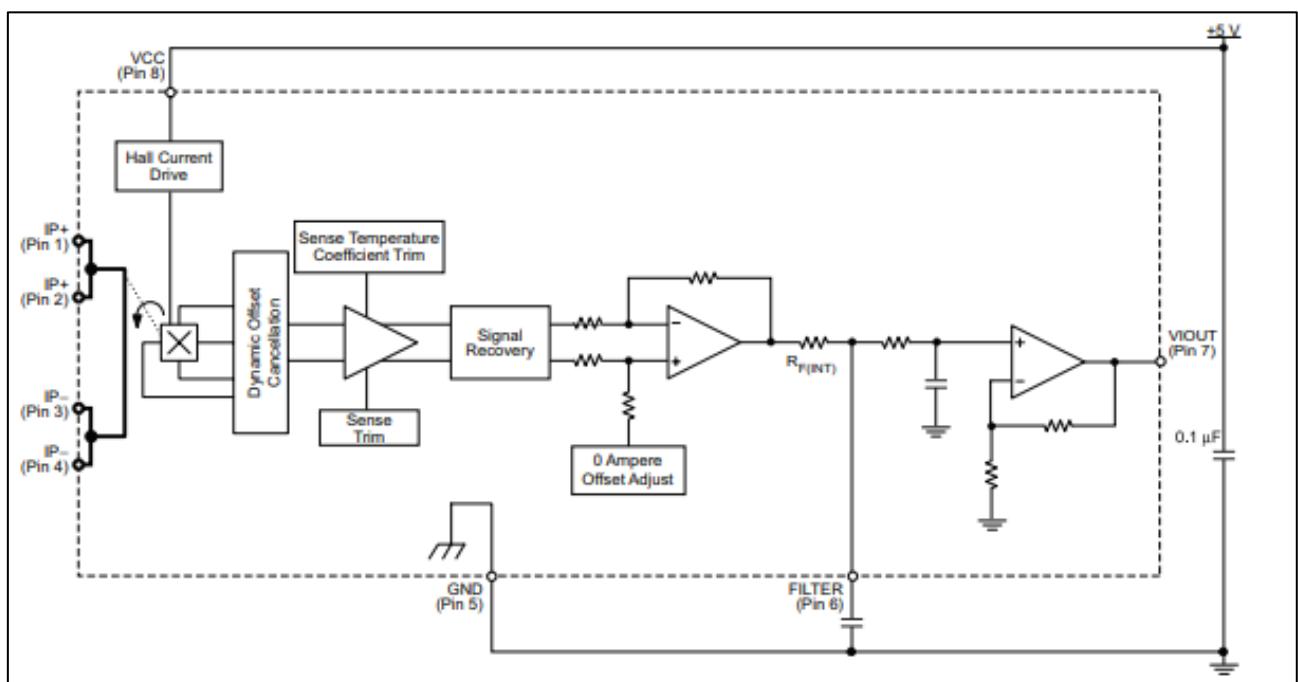
برای محصول اولیه ، از مبدل های سوئیچینگ MORNSUN استفاده شد که با توجه به قیمت تمام شده ی آن ، برای محصول نهایی ، یک آداپتور آماده تهیه شد و در باکس قرار گرفت.

## ۲-۳ طراحی سخت افزار

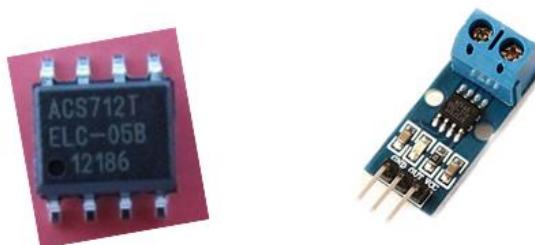
در این بخش در ابتدا به بررسی عملکرد سنسور ها پرداخته شد و در انتهای نقشه‌ی شماتیک و طراحی PCB بررسی شد.

### -۱-۲-۳ حسگر های استفاده شده در مدار :

(۱) ACS712 : با توجه به دیتاشیت قطعه اندکی به جزئیات داخلی آن می‌پردازیم ، این قطعه به صورت سری با ورودی AC مدار قرار می‌گیرد ، جریان از PIN1,2 وارد و از PIN3,4 خارج می‌شود ، عبور جریان از این بخش باعث ایجاد ولتاژ اندکی در بخش مشخص شده با ضرب در می‌شود. سپس با فیلتر کردن و تقویت آن در خروجی (PIN7) قرار می‌گیرد.[7]

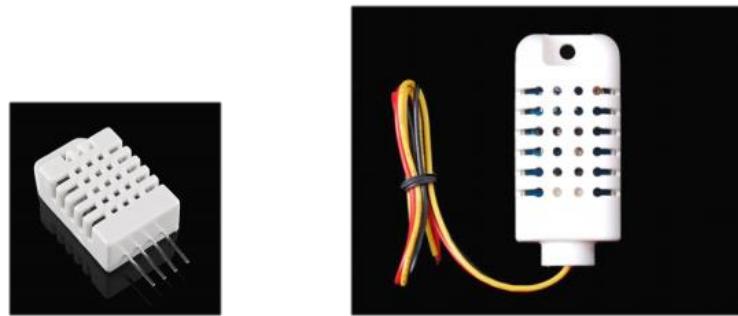


شکل ۴-۳ جزئیات بیشتری از حسگر هال

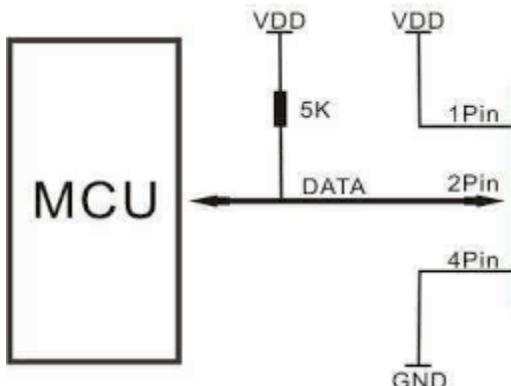


شکل ۵-۳ مازول و حسگر هال

سنسور رطوبت DHT22، یک سنسور عالی جهت سنجش دما و رطوبت هواست، این سنسور از دو سنسور تشکیل شده است ، یک سنسور رطوبت خازنی و یک مقاومت گرمایی. یک تراشه بسیار ساده نیز درون آن وجود دارد که تبدیل آنalog به Digital را انجام می دهد و خروجی Digital را تولید می کند.



شکل ۶-۳ حسگر DHT22

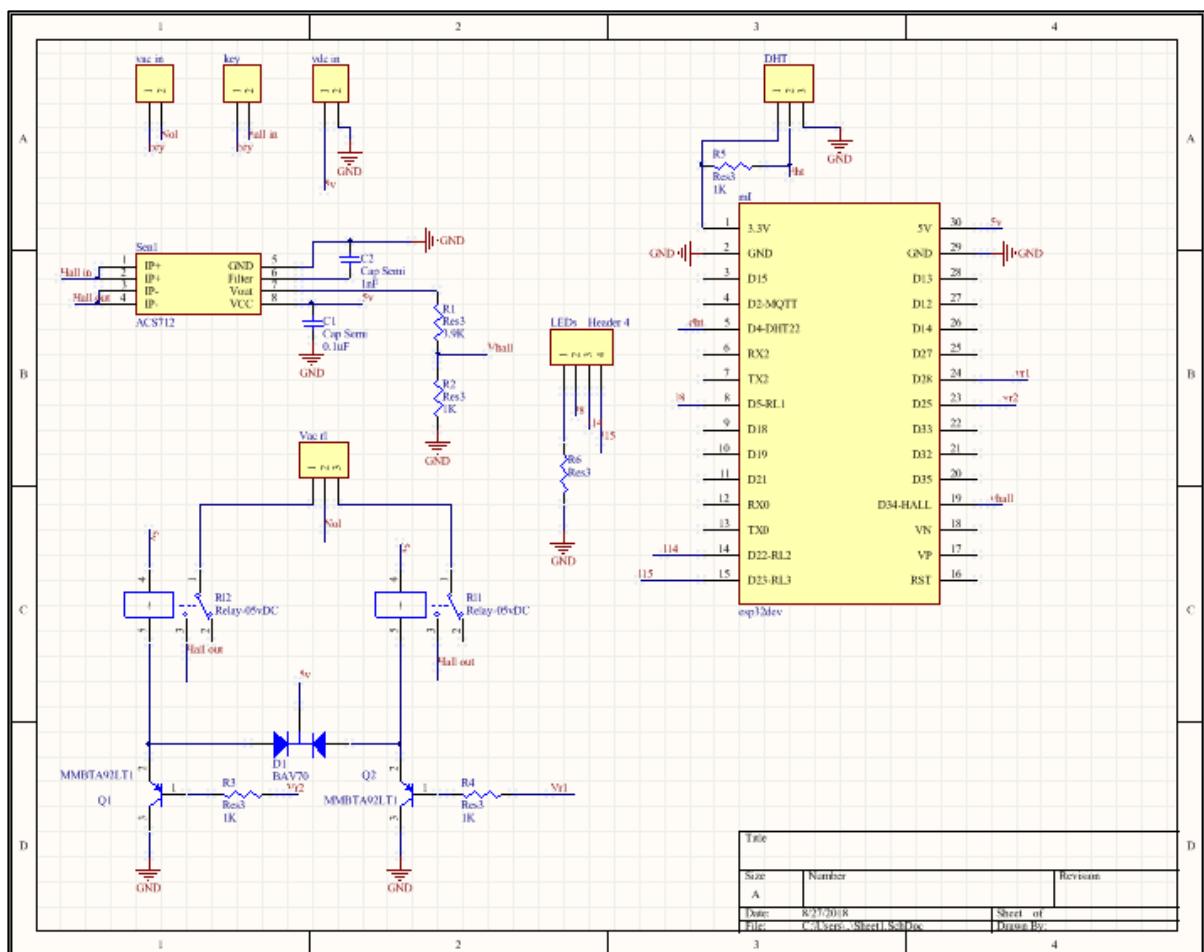


با توجه به دیتا شیت ، نحوه قرار گرفتن این سنسور در مدار مانند شکل ۷-۳ است.[8]

شکل ۷-۳ نحوه اتصال میکروکنترلر و حسگر

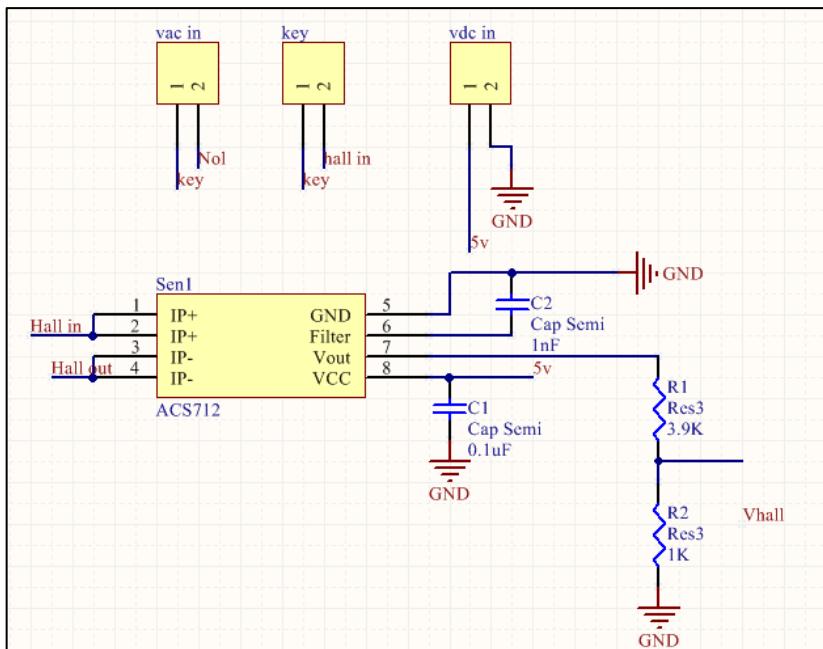
-۲-۲-۳

## نقشه‌ی شماتیک محصول نهایی



شکل ۸-۳ شماتیک محصول نهایی

حال به بررسی شماتیک محصول نهایی و توضیح هر یک از بخش‌های فوق پرداخته می‌شود.

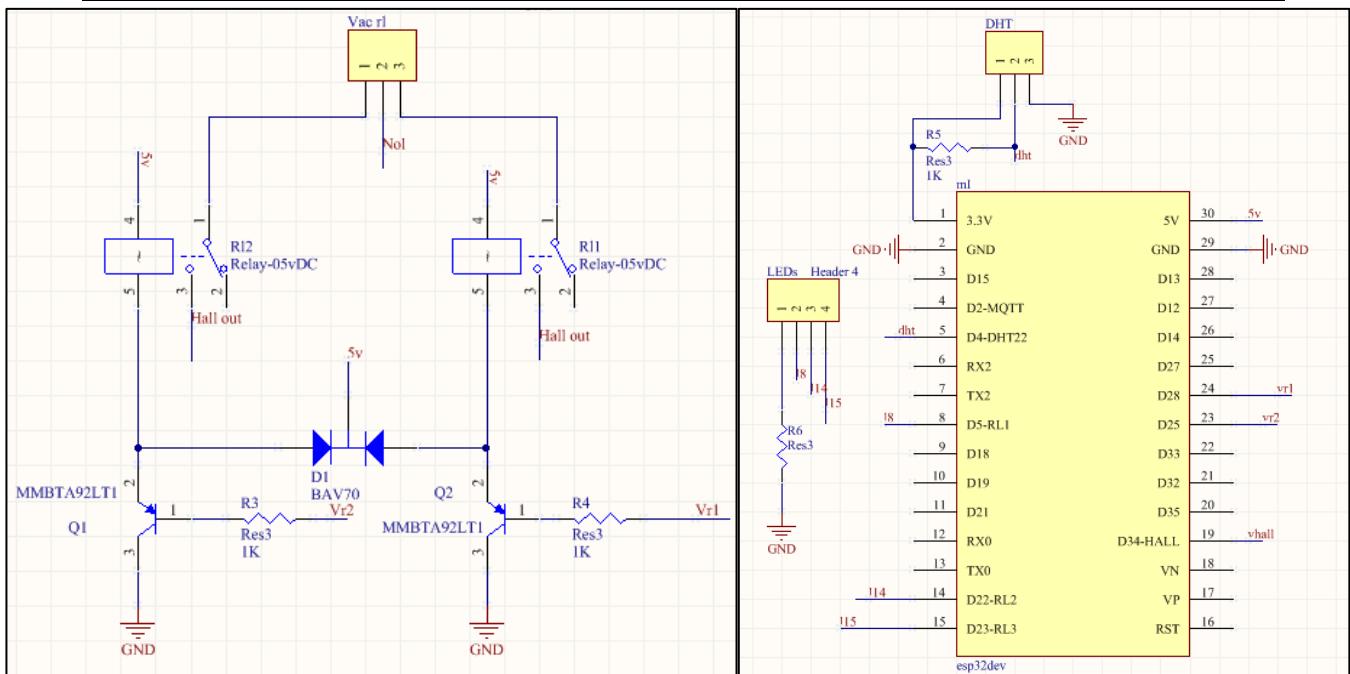


شکل ۹-۳ بخشی از شماتیک محصول نهایی

در شکل فوق دو ورودی AC (vac in) و DC (key) مدار مشاهده می‌شود که ، ورودی AC ابتدا به کلید (key) رفته و بعد از آن به حسگر هال(Sen1) می‌رود. ورودی DC مقدار ۵ ولت دارد که از آن به عنوان تغذیه سنسور استفاده می‌شود. خروجی سنسور در بازه ۰,۵ تا ۴,۵ ولت قرار دارد ولی ADC میکرو تنها بازه ۰ تا ۱ ولت را میتواند قرائت کند، پس خروجی آن را با یک تقسیم مقاومتی به ADC میکرو کنترلر می‌دهیم.

در شکل ۱۱-۳ عملگر روشن/خاموش را مشاهده می‌شود که در آن از دو ترانزیستور MMBTA که پکیج SMD دارند دو رله را کنترل(درایو) می‌کنند. همچنین دیود Bav70 برای محافظت از ترانزیستورها برای تخلیه انرژی ذخیره شده در سلف رله ، به هنگام قطع شدن رله ها استفاده شده است. و کانتور Vac rl برای خروجی استفاده می‌شود که در انتهای رله ، پربیز روی باکس متصل می‌شود.

علاوه بر آن خروجی ولتاژ DC را توسط یک تنظیم کننده ولتاژ ، به میکروکنترلر می‌دهیم.



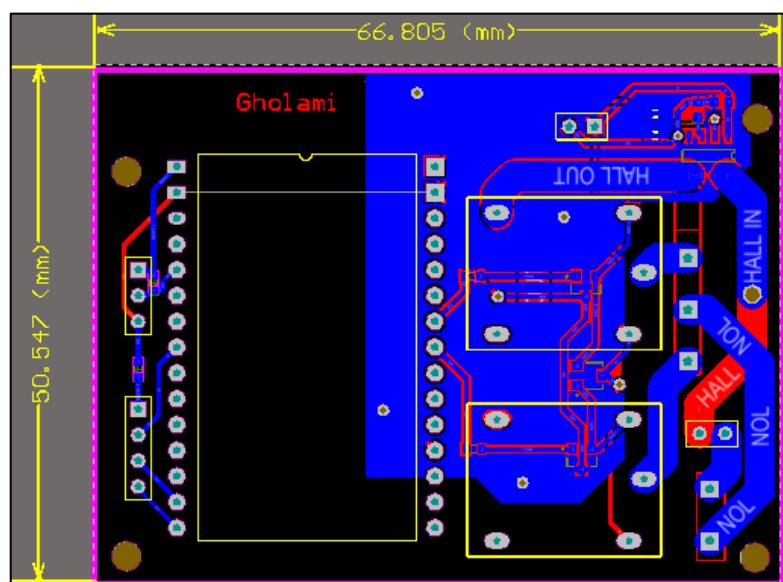
شکل ۱۱-۳ بخشی از شماتیک محصول نهایی

شکل ۱۰-۳ بخشی از شماتیک محصول نهایی

در شکل ۱۰-۳ میکروکنترلر ESP32 را مشاهده می کنیم که توسط کانکتور DHT ، حسگر DHT22 که بیرون از جعبه قرار می گیرد به میکروکنترلر متصل می شود. و توسط کانکتور LEDs الایدی های رو جعبه توسط میکروکنترلر ، مدیریت می شوند.

### ۳-۲-۳ - طراحی فیبر مدار چاپی

پس طراحی شماتیک محصول نهایی ، نوبت به طراحی فیبر مدار چاپی میرسد.



شکل ۱۲-۳ فیبر مدار چاپی محصول نهایی

طراحی این بخش فوق توسط نرم افزار Altium Designer انجام شد. همچنین برای بهبود عملکرد مدار، به جهت نویز گیری در بعضی قسمت های آن Polygon pour کشیده شد. قطر track های جریان بالا (ولتاژ ac) ۳ میلی متر و قطر track های جریان پایین(ولتاژ dc) ۰,۵ میلی متر در نظر گرفته شد. همانطور که در شکل ۱۲-۳ مشاهده می کنید برد دو رو، می باشد و براساس سفارش، طرح فوق با روکش سبز و ماده راجرز چاپ شد.

### ۳-۳-۳ طراحی نرم افزار

در این بخش به طراحی نرم افزار اشاره می شود ، در ابتدا جریان اجرای برنامه درون ESP32 را مرور کرده و در ادامه پروتکل MQTT را معرفی نموده و در پایان ، نحوه خواندن سنسور ها و فلوچارت کد زده شده، بررسی خواهد شد.

#### ۳-۳-۱-۱ جریان اجرای برنامه [9]

همانطور که در بخش سخت افزار توضیح داده شد ESP32 دارای ۲ پردازنده می باشد که با نام های زیر نام گذاری می شوند: CPU-PRO,CPU-APP . در این بخش به توضیح اتفاقاتی که قبل از اجرای main() رخ می دهد و فرایند ها و اتفاقاتی که این دو پردازنده انجام می دهند، شرح داده می شود.

به طور کلی سه مرحله تا اجرا شدن main وجود دارد که به توضیح هر کدام پرداخته می شود:

۱) بعد از فرمان شروع مجدد<sup>۱</sup> ، CPU PRO بلافاصله شروع به فعالیت می کند و اقدام به اجرای بردar شروع مجدد که در خانه ۰x40000400 رام<sup>۲</sup> که قابل تغییر نیست، را می کند. در همین هنگام CPU APP در

<sup>۱</sup> Reset

<sup>۲</sup> ROM

حال شروع مجدد است. PRO CPU همه‌ی آمادگی‌سازی‌ها<sup>۱</sup> را انجام می‌دهد و در نهایت اعلام آماده سازی به APP CPU انتقال می‌دهد.

(۲) در این مرحله APP CPU بعد شروع مجدد خود و دریافت سیگنال آماده سازی PRO CPU در خواستی را به آن ارسال می‌کند. تا برنامه را دریافت کند. بعد از این سیگنال PRO CPU داده‌هایی که مربوط به کد کامپایل شده و ذخیره شده‌ی ما می‌باشد را به APP CPU انتقال می‌دهد.

حال اینجا دو مسئله مطرح می‌شود که به تبیین آن پرداخته می‌شود :

مسئله اول : این که کد کامپایل شده ما در کجا قرار گرفته است ؟

مسئله دوم : چه دلیلی دارد که در دو مرحله اطلاعات را در اختیار CPU قرار دهیم؟

پیرامون مسئله اول می‌توان گفت که کد کامپایل شده ما در حافظه‌ای به نام ROM قرار دارد که با شروع کاری PRO CPU آن را به IRAM (INSTRUCTION RAM) انتقال می‌دهد و کدها از آن جا شروع به عملکردن می‌کنند. پس اولین وظیفه PRO CPU انتقال اطلاعات برنامه ما است.

پیرامون مسئله دوم ، واقعاً برای خواننده این سوال پیش می‌آید که چرا می‌بایست همچنین روندی طی شود ، جواب آن این است که؛ در این نوع معماری و ساختار به میکروکنترلر این اجازه داده می‌شود که بدون

اینکه اتصال فیزیکی برقرار شود بتوان کد جدیدی را روی آن منتقل نمود که با<sup>۲</sup> OTA شناخته می‌شود.

که جزو پرکاربرد ترین قابلیت‌های است. همچنین می‌توان کد انتقال یافته روی ROM را با الگوریتم‌های تغییر داد و امنیت آن را افزایش داد و با پردازش توسط PRO CPU به حالت اولیه بازگرداند ، تا امکان مهندسی معکوس ، و سرقت اطلاعات میکرو کنترلر را از بین برد.

<sup>1</sup> initialization

<sup>2</sup> Over the Air

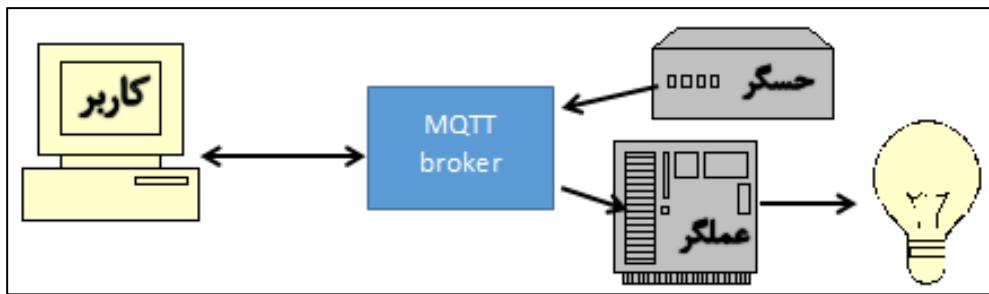
۳) در اخرين مرحله ( main\_task ) اجرا مي شود ، در پروژه فعلی به علت پيچيده نبودن مسئله تنها از APP CPU استفاده شد. پس برای اين امر تنها تابع ( main\_app ) صدا شد. و برای تمامی پردازش هاي هنگام کار از اين پردازنده بهره برده شد ، وهمچنین در صورت توسعه پروژه می توان OTA و امنیت آن را با توجه به راه کار ياد شده در بالا بهبود بخشید ، البته در صورت پيچيده بودن پردازش مسئله می توان از پردازنده ديگر که PRO CPU نام دارد برای بعضی کاربرد ها استفاده کرد، مثلا در صورتی که تعداد سنسور ها زیاد یا حجم پردازش زیاد باشد می توان از پردازش های موازی بهره برد . و از ( main\_pro ) نیز بهره برد و راه ارتباطی بین پردازنده نیز ( Data Ram ) DRAM استفاده کرد.

### -۲-۳-۳ MQTT<sup>۱</sup> چیست؟ [10]

MQTT یک پروتکل سبک انتقال اطلاعات صفت بندی شده است. همانطور که از نام آن پیداست (انتقال از راه دور پیام های صفت بندی شده) برای ارسال اطلاعات از راه دور مناسب است. از آن جایی که بسیار سبک و ساده است بنابراین برای سناریو های WSN، M2M، IOT که سنسور ها و عملگر ها از طریق یک MQTT broker باهم به تبادل اطلاعات می پردازند بسیار کاربردی است. مثال :

- \* یک سنسور نور به صورت زمان پیوسته داده های خود را به سمت broker ارسال می کند .
- \* ساختن یک نرم افزار که داده ها را از broker دریافت کند و تصمیم بگیرد که آیا نور مناسب است یا خیر و ارسال فرمان به broker
- \* ساختن یک عملگر که داده ها ( فرمان ها ) را از broker دریافت کند و اقدام عملی محیط پیرامون که روشن و خاموش کردن چراغ های محیط است را انجام دهد.

<sup>۱</sup> Message Queuing Telemetry Transport



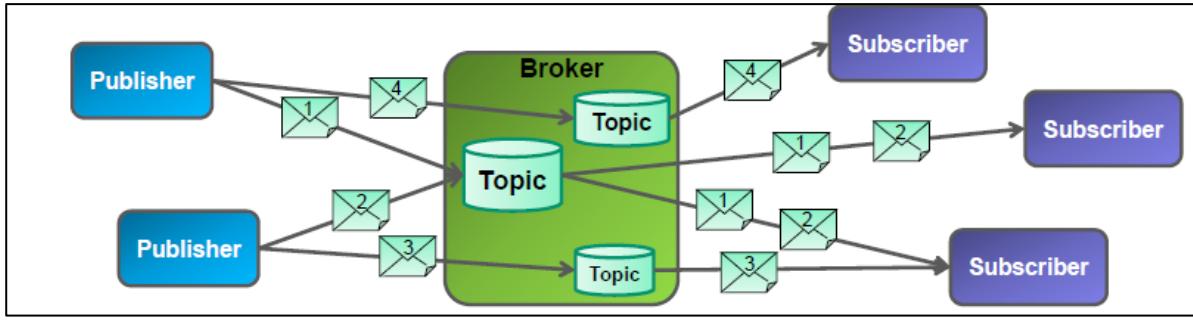
شکل ۳-۱۳ نحوه اتصال کاربران سرور MQTT

## ۱-۲-۳-۴- ویژگی های کلیدی MQTT

اولین و شاید مهمترین ویژگی MQTT را می‌توان سبک(ساده) بودن پیام‌ها و صفت‌بندی شده و قوانین انتقال آن را در نظر گرفت. ویژگی بعدی را می‌توان مخابره آسنکرون اطلاعات در نظر گرفت، که به صورت رویداد<sup>۱</sup> محور مخابرها انجام می‌شود. که این امکان را می‌دهد بعضی امکانات را غیرفعال کرده تا رویداد جدید ظاهر شود، و با رخداد یک رویداد، فرایندی مشخص را انجام داد. ویژگی کلیدی دیگری که می‌توان در نظر گرفت، این است که پهنهای باند کمی را اشغال می‌کند ( تنها ۲٪ بایت هدر به پکت‌های داده اضافه می‌کند) که برای شبکه‌ها با پهنهای باند پایین بسیار مناسب است. از دیگر مزیت‌های استفاده از MQTT استفاده از مدل PubSub است که از ساده‌ترین روش‌های انتقال اطلاعات است. به نحوی که ارسال کننده، داده‌ها را Publish و شنونده، داده‌ها را Subscribe می‌کند. از ویژگی‌های اساسی این پروتکل، جدا کردن اطلاعات فرستنده، شنونده‌های مختلف از طریق topic هاست. به نحوی که مثلاً فرستنده شماره ۱ داده‌های خود را در تاپیک A و فرستنده شماره ۲ داده‌های خود در تاپیک B ارسال می‌کند و شنونده ۱ داده‌های خود را از تاپیک A و شنونده ۲ داده‌های خود را از تاپیک B دریافت می‌کند. یکی دیگر از ویژگی‌ها این است که ارتباط فرستنده و شنونده با broker از طریق پروتکل TCP<sup>۲</sup> است، که ارسال داده و امنیت آن را (نسبت به UDP) تضمین می‌کند.

1 event

## <sup>2</sup> Transmission Control Protocol

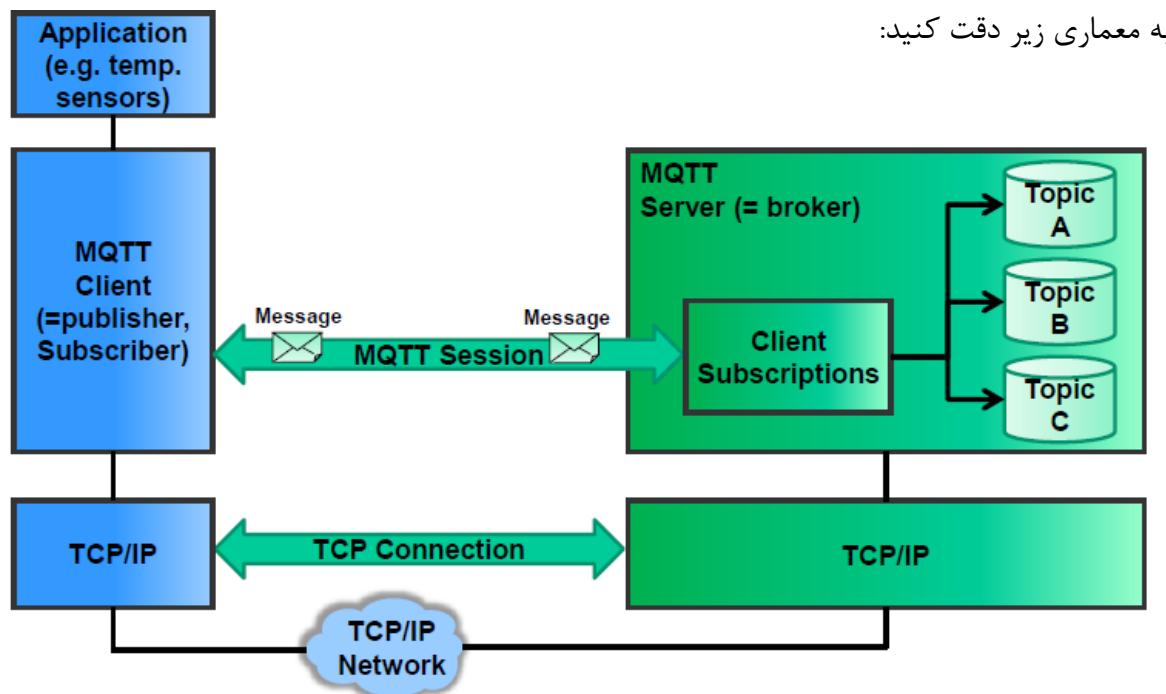


شکل ۳-۱۴ نحوه تبادل اطلاعات بین کاربران

### ۲-۲-۳-۳ مدل MQTT

بخش های اصلی یک ارتباط از طریق MQTT شامل :

(۱) کاربران<sup>۱</sup> (فرستنده و شنونده) (۲) سرور (broker) (۳) جلسه<sup>۲</sup> (۴) تاپیکها (۵) اشتراک<sup>۳</sup>



شکل ۳-۱۵ روند کلی پروتکل

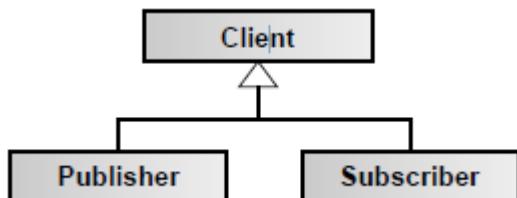
<sup>۱</sup> MQTT client

<sup>۲</sup> sessions

<sup>۳</sup> subscription

### ۳-۲-۳-۳- کاربران MQTT

کاربران MQTT ، شامل فرستنده‌گان(Publisher) و مشترک‌شوندگان(Subscriber) می‌شوند، آن‌ها می‌بایست در یک تاپیک مشترک شوند (subscribe کنند) تا بتوانند داده را ارسال و دریافت کنند.



شکل ۱۶-۳ کاربران

### ۳-۲-۳-۴- سرور (broker) MQTT

سرور‌ها تاپیک‌ها را می‌سازند و اجرا می‌کنند. یعنی هنگامی که کاربران (فرستنده‌ها و شنونده‌ها) روی یک تاپیک مشترک می‌شوند، (subscribe می‌کنند) ، وظیفه دریافت داده‌ها و انتقال آن را بر عهده دارند.

### ۳-۲-۳-۵- تاپیک‌ها

به طور دقیق می‌توان گفت تاپیک‌ها محل تشکیل صفات داده‌ها هستند، البته در پروژه فعلی به علت پایین بودن تعداد داده‌ی تبادل شده، صفات تشکیل نمی‌شود و داده‌ها به سادگی پردازش می‌شوند، ولی به توضیح آن پرداخته شد. تاپیک‌ها به کاربران این اجازه را می‌دهند تا داده‌های خود را تعریف و معنا دهند.

به طور مثال تعدادی از تاپیک‌های نام‌گذاری شده در پروژه که به کار و نوع داده‌های درون تاپیک مفهوم /sensors/temp می‌بخشد :

/sensors/humid

/command

### ۳-۲-۳-۶- جلسه (session)

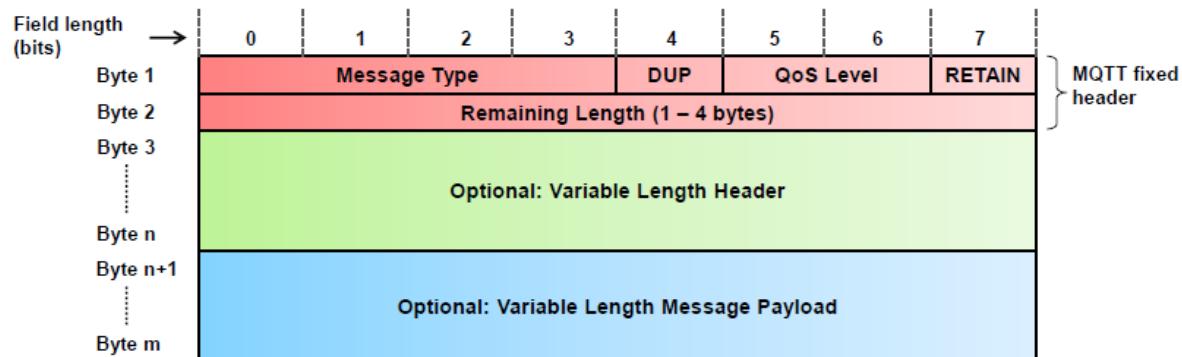
یک session ، معرفی کننده یک کاربر MQTT به سرور است. تمامی ارتباطات بین کاربران و سرور از طریق session انجام می‌شوند، در بخش ۳-۵ با ذکر مثال توضیحات بیشتری ارائه می‌گردد.

### ۳-۳-۲-۷- اشتراک(subscription)

برخلاف جلسه(session) ، اشتراک ،یک کاربر را به یک تاپیک متصل می‌کند ، و هنگامی که کاربر با یک تاپیک مشترک (subscribed) شد ، می‌تواند داده‌های درون تاپیک را تغییر دهد. در بخش ۵-۳ توضیحات بیشتری ارائه می‌گردد.

### ۳-۳-۲-۸- ساختار پیام‌ها در پروتکل MQTT

به طور کلی پیام‌ها در این پروتکل از شکل زیر (شکل ۳-۵) پیروی می‌کنند و با تغییر نوع پیام ساختار‌ها کمی تغییر می‌کنند ، در این بخش به صورت مختصر با برخی از جزئیات این پروتکل آشنا می‌شویم.



Message fixed header field	Description / Values	
Message Type	0: Reserved	8: SUBSCRIBE
	1: CONNECT	9: SUBACK
	2: CONNACK	10: UNSUBSCRIBE
	3: PUBLISH	11: UNSUBACK
	4: PUBACK	12: PINGREQ
	5: PUBREC	13: PINGRESP
	6: PUBREL	14: DISCONNECT
	7: PUBCOMP	15: Reserved
DUP	Duplicate message flag. Indicates to the receiver that this message may have already been received. 1: Client or server (broker) re-delivers a PUBLISH, PUBREL, SUBSCRIBE or UNSUBSCRIBE message (duplicate message).	
QoS Level	Indicates the level of delivery assurance of a PUBLISH message. 0: At-most-once delivery, no guarantees, «Fire and Forget». 1: At-least-once delivery, acknowledged delivery. 2: Exactly-once delivery. Further details see <a href="#">MQTT QoS</a> .	
RETAIN	1: Instructs the server to retain the last received PUBLISH message and deliver it as a first message to new subscriptions. Further details see <a href="#">RETAIN (keep last message)</a> .	
Remaining Length	Indicates the number of remaining bytes in the message, i.e. the length of the (optional) variable length header and (optional) payload. Further details see <a href="#">Remaining length (RL)</a> .	

شکل ۳-۱۷ فریم کلی در پروتکل MQTT

همانطور که در دو شکل فوق دیده می‌شود به طور خلاصه:

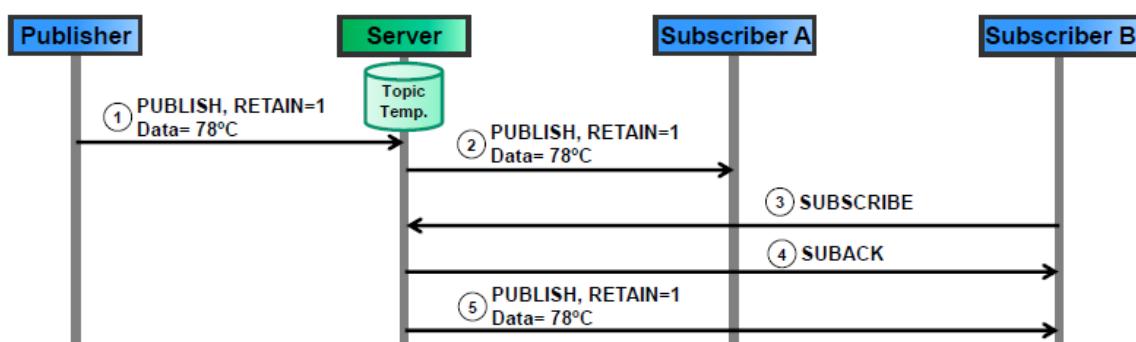
۱۴ : نوع متفاوت پیام منتقل میشود (۱۵ تا ۱) که در بخش بعدی اندکی با

جزئیات آن آشنا می‌شویم. البته نوع فریم بندی دقیق آن در فصل ششم آورده شد.

<sup>۱</sup> این پرچم نشان می‌دهد که آیا پیام ارسال شده تکراری است؟ یا خیر.

• RETAIN: اگر این پرچم ۱ باشد ، سرور موظف است اخرين پیام را نگه دارد و در صورت اشتراك

کاربر جديد اطلاعات ذخیره شده را به آن ارسال کند .



شکل ۱۸-۳ نحوه تبادل اطلاعات در پروتکل MQTT

• remaining length: تعداد بایت های پیامی که منتقل خواهد شد را در بر خواهد داشت.

• QoS<sup>۲</sup>: MQTT به مقصد رساندن داده ها را تضمین می کند، با وجود اينکه TCP/IP به مقصد

رساندن اطلاعات را تضمین می کند ولی اگر مسیر ارتباطی TCP قطع شود ، داده ها ممکن است

از بين بروند، بنابراین MQTT سه سطح کیفت از سرویس را ارائه می دهد.

○ QoS level 0: حداکثر یک بار ارسال: در این سطح تضمین همان ، تضمین TCP/IP است

و اطلاعات تنها یک بار ارسال می شوند؛ کاربرد: مثلا اطلاعات سنسور دما ، چرا که اگر

یکی از اطلاعات را از دست دهیم برایمان اهمیتی چندانی ندارد ، چرا که میزان متوسط

آن برایمان اهمیت دارد، و اگر داده تکراری داشته باشیم ، متوسطگیری غلط خواهد بود.

<sup>1</sup> Duplicate Message Flag

<sup>2</sup> Quality of Service

○ QoS level 1 : حداقل یک بار ارسال : در این سطح تضمین دریافت داده ها قطعی است

ولی داده اضافی ایجاد می شود. کاربرد : مثلا سنسور درب منزل باز و بسته بودن درب را

گزارش می دهد ولی برای ما تغییرات در حالت درب اهمیت دارد و داده تکراری مهم

نیست و همچنین نباید این داده از بین برود، چرا که در عملگر شاید یک دزدگیر

فعال شود.

○ QoS level 2 : دقیقا یک بار ارسال : این بالاترین کیفیت ارائه شده است که تضمین

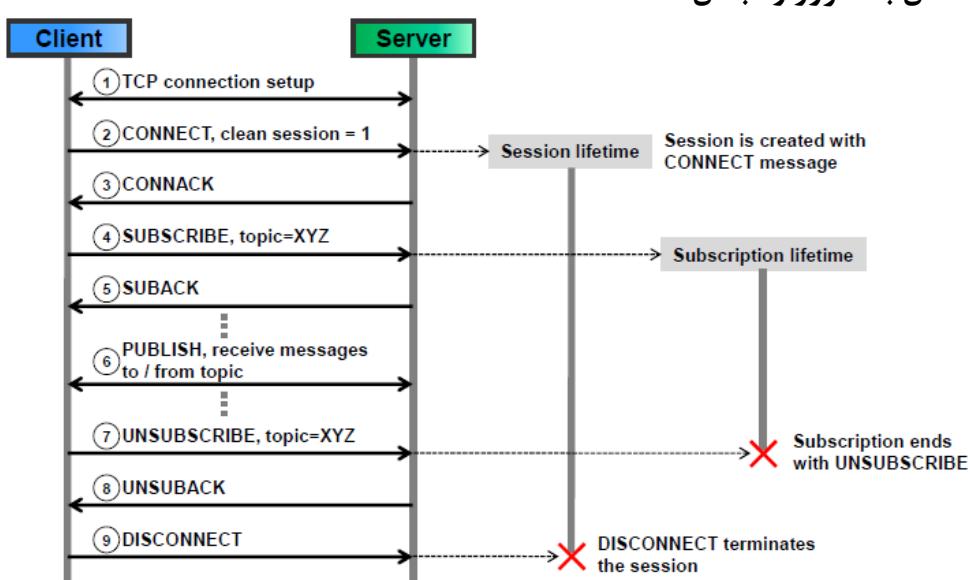
می کند داده دقیقا یک بار به مقصد می رسد ، البته برای این سطح از کیفیت باید پیام را

ذخیره کرد. کاربرد ؛ در محیط هایی که وجود اطلاعات اضافی باعث ایجاد اخلال

می شود. مثلا؛ به صدا در آمدن یک صدا به عنوان عملگر که دقیقا باید یک بار پیام به مقصد

بررسد.

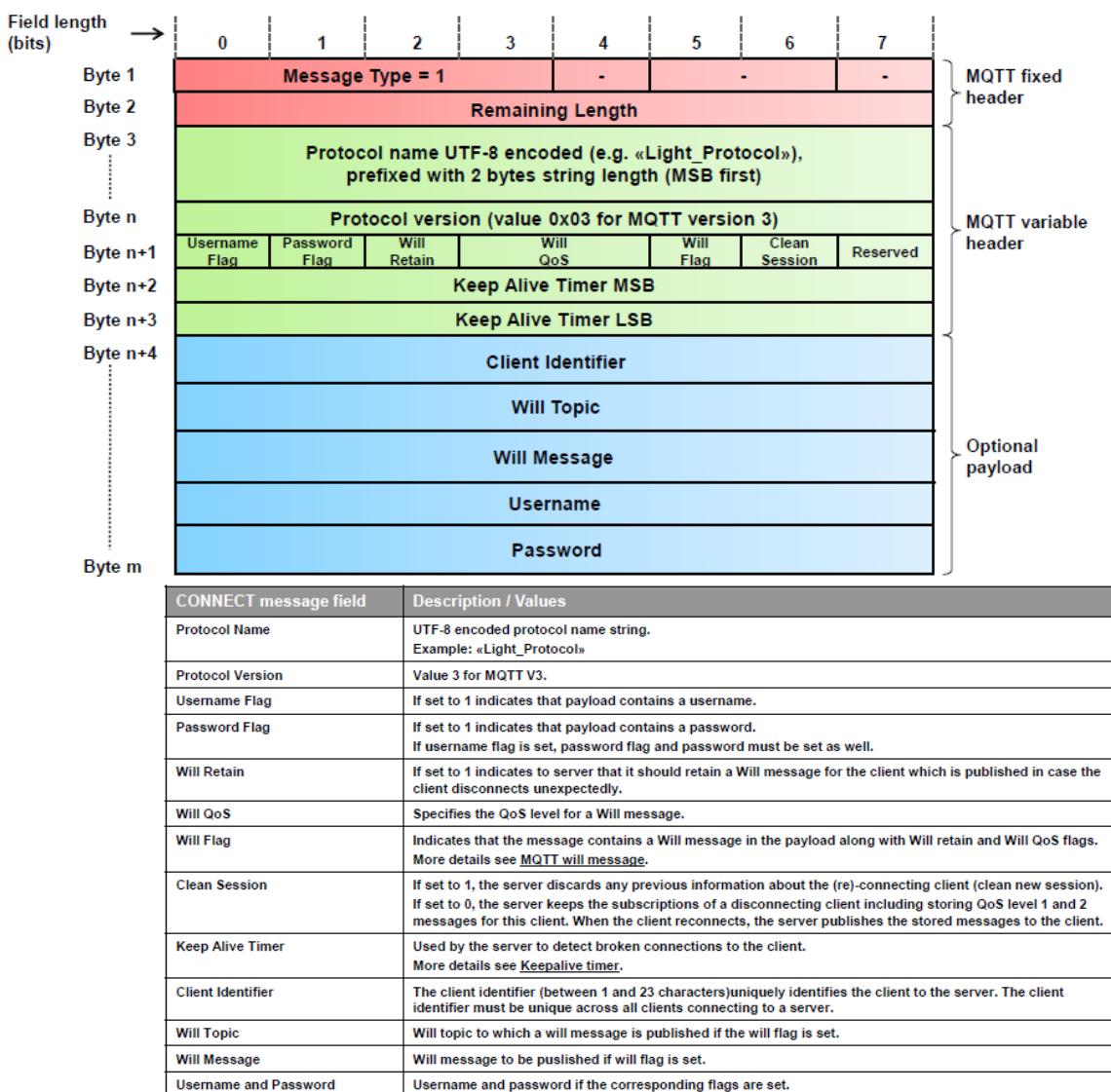
### - ۳-۲-۹ - اتصال به سرور و تبادل داده ها



شكل ۱۹-۳ انواع مختلف پیام بین کاربر و سرور

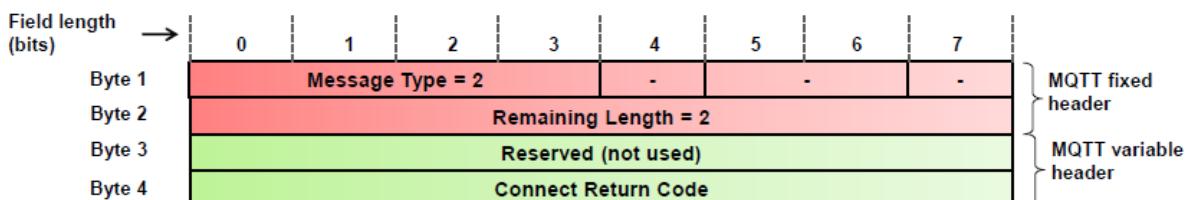
۱) برای اینکه کاربر(فرستنده و شنونده) بتوانند به سرور متصل شوند باید یک اتصال TCP\IP داشته باشند

۲) در مرحله بعدی مناسب با کاربردی که دارند دستور CONNECT را با فریم زیر ارسال کنند:



شکل ۲۰-۳ مدل پیام ۱ در پروتکل MQTT

۳) اگر اطلاعات وارد شده صحیح باشد و کاربر بتواند با سرور ارتباط برقرار کند ، جلسه(Session) برقرار می شود. و در پاسخ CONNAK با فریم زیر ارسال می شود:

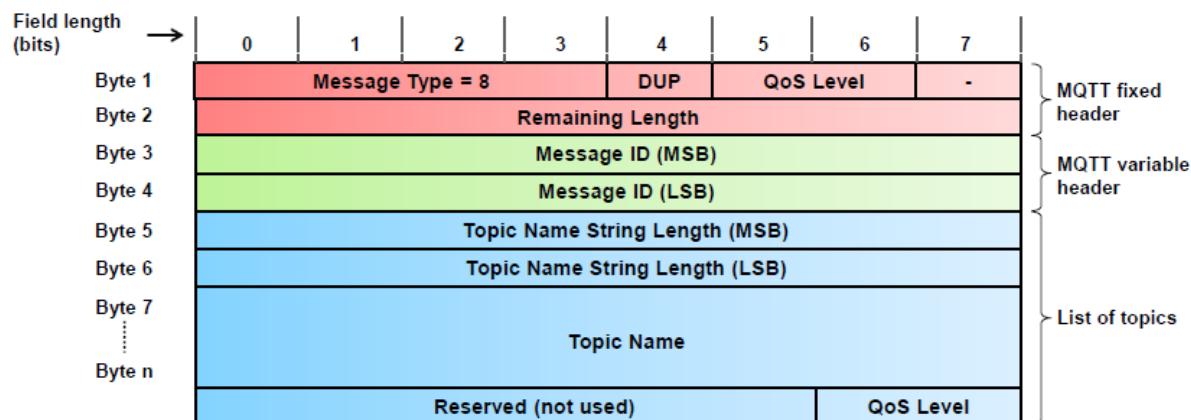


CONNACK message field	Description / Values
Reserved	Reserved field for future use.
Connect Return Code	0: Connection Accepted 1: Connection Refused, reason = unacceptable protocol version 2: Connection Refused, reason = identifier rejected 3: Connection Refused, reason = server unavailable 4: Connection Refused, reason = bad user name or password 5: Connection Refused, reason = not authorized 6-255: Reserved for future use

شکل ۲۱-۳ مدل پیام ۲ در پروتکل MQTT

۴) حال وقت آن رسیده که کاربر روی یکی از تاپیک ها اشتراک یابد.(subscribe شود).

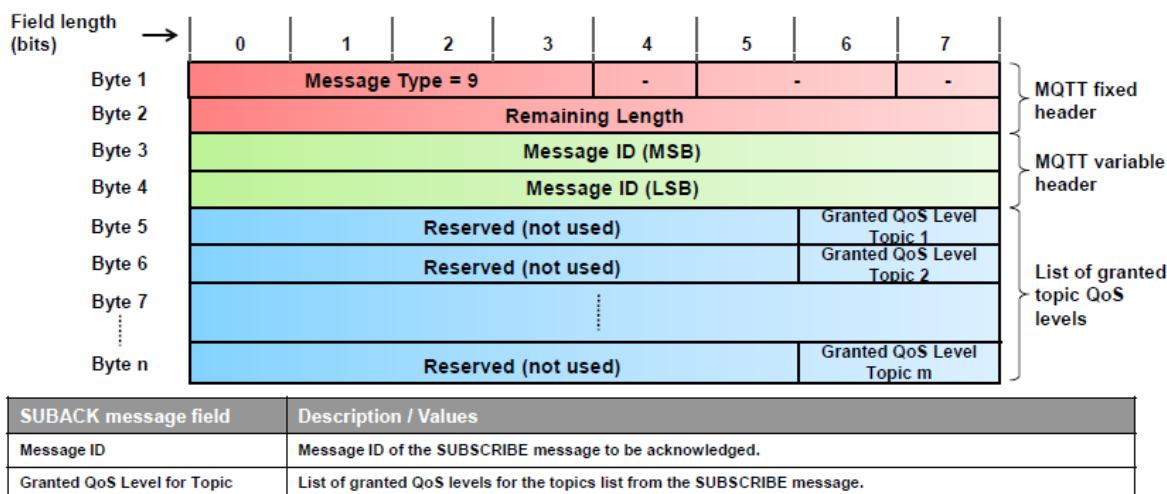
که باید از طریق فریم زیر تاپیکش را مشخص کرده و مشترک شود.



SUBSCRIBE message field	Description / Values
Message ID	The message ID field is used for acknowledgment of the SUBSCRIBE message since these have a QoS level of 1.
Topic Name with Topic Name String Length	Name of topic to which the client subscribes. The first 2 bytes of the topic name field indicate the topic name string length. Topic name strings can contain wildcard characters as explained under <a href="#">Topic wildcards</a> . Multiple topic names along with their requested QoS level may appear in a SUBSCRIBE message.
QoS Level	QoS level at which the clients wants to receive messages from the given topic.

شكل ۳-۲ مدل پیام ۸ در پروتکل MQTT

۵) اگر مرحله قبل موفقیت آمیز باشد، سرور پیامی را با فریم زیر ارسال می کند.



شكل ۳-۳ مدل پیام ۳ در پروتکل MQTT

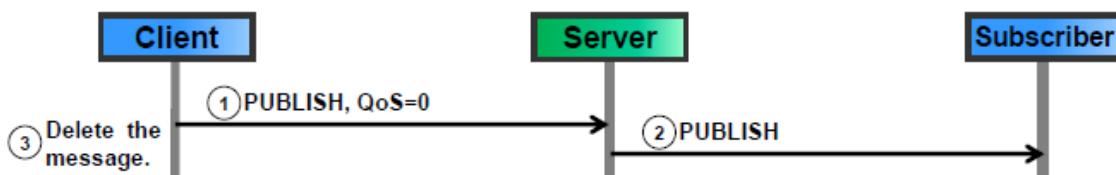
۶) مشابه روند فوق می توان از اشتراک یک تاپیک خارج شد.(unsubscribe کرد).

۷) مجددا سرور به کاربر پیام تایید یا عدم تایید آن را می دهد .

۸) همچنین می توان از سرور قطع شد که پایان جلسه(session) را معنی می دهد.

حال به توضیح مرحله ششم یعنی مرحله ای که جلسه برقرار است و کاربرها را تاپیک ها مشترک هستند ، پرداخته می شود. بسته به اینکه از چه کیفیت سرویسی (QoS0,QoS1,QoS2) استفاده می شود، نحوه ارتباط متفاوت است.

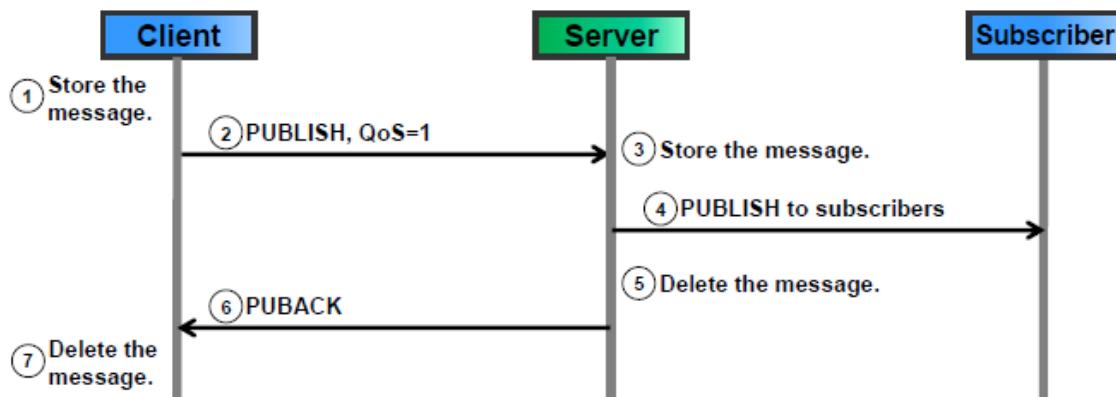
(۱) QoS0 : حداکثر یک بار ارسال



شکل ۲۴-۳ کیفیت سرویس دهی نوع صفر

به این صورت است که کاربر داده ها را به سمت سرور ارسال می کند و سرور هم همان داده را به مشترک ارسال می کند. همچنین بعد از این که کاربر اطلاعات را ارسال کرد ، داده را حذف می کند. در اصطلاح به آن fire and forget می گویند.

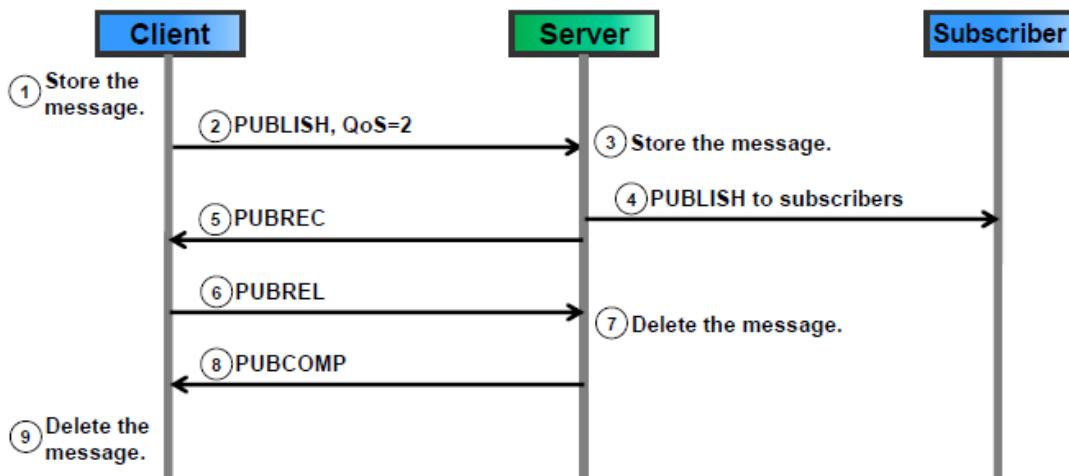
(۲) QoS1 : حداقل یک بار ارسال



شکل ۲۵-۳ کیفیت سرویس دهی نوع یک

به این صورت است که ابتدا داده ارسالی ذخیره می شود ، بعد از آن به سمت سرور ارسال می شود، سرور اطلاعات را در خود ذخیره کرده و به سمت مشترک ارسال می کند، و بعد از آن از سرور داده ها را پاک می کند و نتیجه ارسالش را به کاربر می فرستد ، کاربر با دانستن آگاهی از رسیدن داده ها به مقصد داده ذخیره شده را حذف می کند.

### ۳) QoS2: دقیقاً یک بار

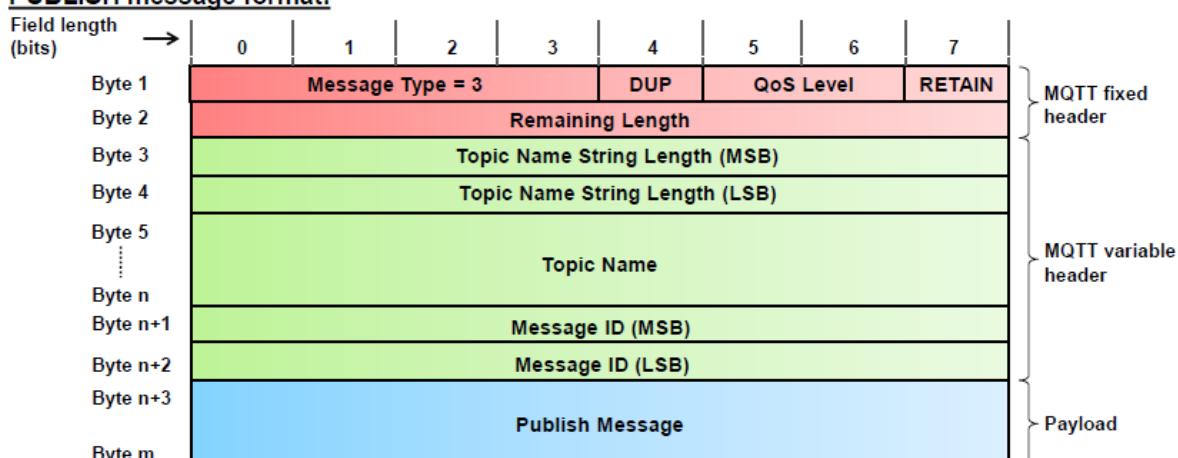


شکل ۲۶-۳ کیفیت سرویس دهی نوع دو

در این روش دقیقاً یک بار داده ارسال می‌شود، به این نحو که، ابتدا داده‌ها ذخیره می‌شود. پس از آن به سمت سرور ارسال می‌شود، در سرور ذخیره شده و به مشترک ارسال می‌شود، و بعد از ارسال به کاربر پیام ارسال نیز برای او فرستاده می‌شود، آنگاه کاربر، پیامی را به سرور ارسال می‌کند، سرور با دریافت این پیام، داده را پاک می‌کند، و پیغام پاک کردن آن را به کاربر ارسال می‌کند، کاربر هم با دریافت این پیام، داده را حذف می‌کند.

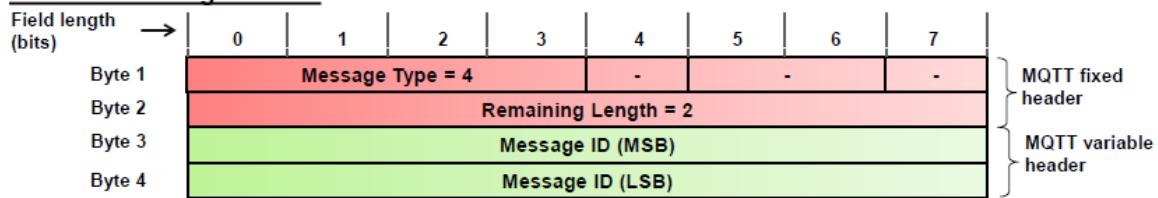
- در ادامه نحوه فریم بندی دستورات مربوط به ارسال اطلاعات قرار داده شد.

#### PUBLISH message format:

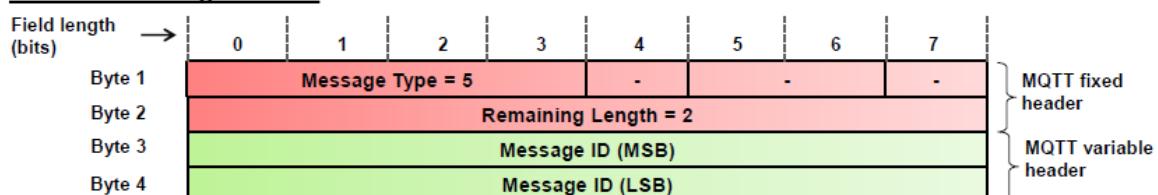


PUBLISH message field	Description / Values
Topic Name with Topic Name String Length	Name of topic to which the message is published. The first 2 bytes of the topic name field indicate the topic name string length.
Message ID	A message ID is present if QoS is 1 (At-least-once delivery, acknowledged delivery) or 2 (Exactly-once delivery).
Publish Message	Message as an array of bytes. The structure of the publish message is application-specific.

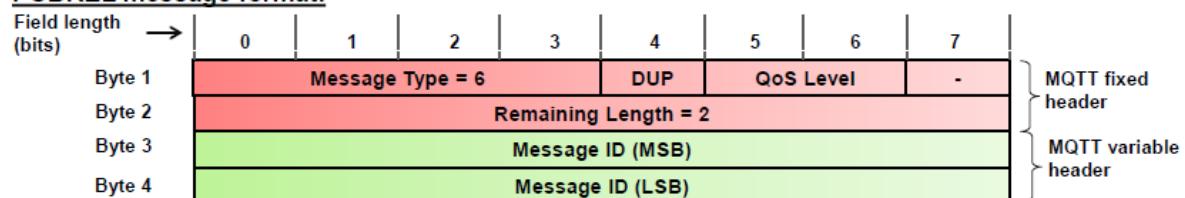
شکل ۲۷-۳ مدل پیام ۳ در پروتکل MQTT

**PUBACK message format:**

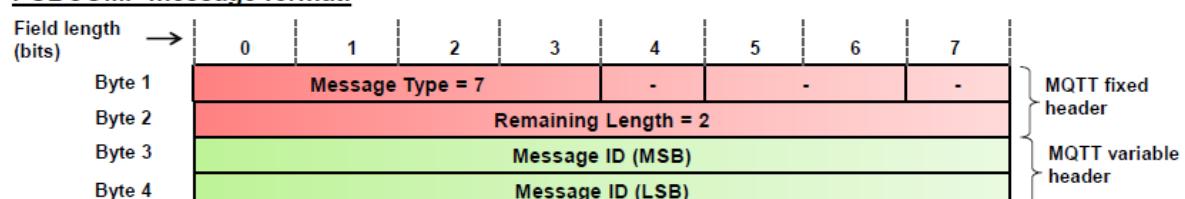
PUBACK message field	Description / Values
Message ID	The message ID of the PUBLISH message to be acknowledged.

**PUBREC message format:**

PUBREC message field	Description / Values
Message ID	The message ID of the PUBLISH message to be acknowledged.

**PUBREL message format:**

PUBREL message field	Description / Values
Message ID	The message ID of the PUBLISH message to be acknowledged.

**PUBCOMP message format:**

PUBCOMP message field	Description / Values
Message ID	The message ID of the PUBLISH message to be acknowledged.

شکل ۲۸-۳ مدل پیام ۴،۵،۶،۷ در پروتکل MQTT

### -۳-۳-۳- انتخاب بروکر مناسب

در فصل سوم به بررسی پروتکل MQTT پرداخته شد و در آن جا سرور یا بروکر و عملکرد آن معرفی شد ، تنها محدودیتی که در انتخاب یک سرور داریم ، این است که ، آیا به ایران سرویس دهی می‌کند یا خیر ، که در این میان :

Google IOT Core- Samsung Artik-Microsoft Azure-Amazon Web Services,...

به ایران سرویس دهی نمی‌کنند ، در این میان علاوه بر امکانات سرویس دهی مسئله‌ها و فاکتور‌های گوناگونی وجود دارد که این بروکر‌ها را از هم متمایز می‌کند برای مثال نوع امنیت در ورود و شناسایی که منجر به ایجاد جلسه(session) می‌شود یا نوع امنیت ارتباط که آیا فردی می‌تواند در میانه راه ارتباط را شنود کند یا اینکه پایگاه داده موجود تا چه زمانی می‌تواند اطلاعات را ذخیره کند. و اینکه آیا به شما API میدهد ؟ تا بتوان در محیط‌های دیگر استفاده نمود . یا اینکه آیا خودش رابط کاربری برای تلفن‌های همراه دارد یا خیر ؟ همه‌ی پلتفرم‌هایی که در فوق ذکر شد این قابلیت‌ها را پشتیبانی می‌کنند ، حتی Artik نیز تا حدود ۲ ماه پیش این امکانات را در اختیار ایرانیان قرار می‌داد ولی به دلیل مسائل سیاسی نشد که از این امکانات بهره ببریم پس به سراغ بروکر‌هایی با امکانات کمتر رفته شد.

که در این میان HiveMQ و cloudmqtt مورد بررسی قرار گرفت و cloudmqtt به دلیل امنیت بیشترش در هنگام متصل شدن مورد استفاده قرار گرفت.

### -۴-۳-۳- طراحی تاپیک‌ها و بستر انتقال داده

با توجه به مدل کسب کار ما دو عملگر خاموش روشن ، یک سنسور دما، یک سنسور رطوبت ، یک سنسور میزان جریان عبوری را به عنوان سخت افزار داریم .

حال قصد داریم تا تاپیک‌ها و کیفیت سرویس دهی (QoS) و نام‌گذاری آن‌ها را در این مرحله مشخص کنیم:

تایپیک ها	کیفیت سرویس	توضیحات
\sensor	\temp	0 میزان دما
	\humid	0 میزان رطوبت
	\hall	0 میزان توان
\relay	\1	1 صفر: روشن 0 یک: خاموش
	\2	1 صفر: روشن 0 یک: خاموش
\command	0	0 خواندن دما
	1	0 خواندن رطوبت
	2	0 خواندن توانی

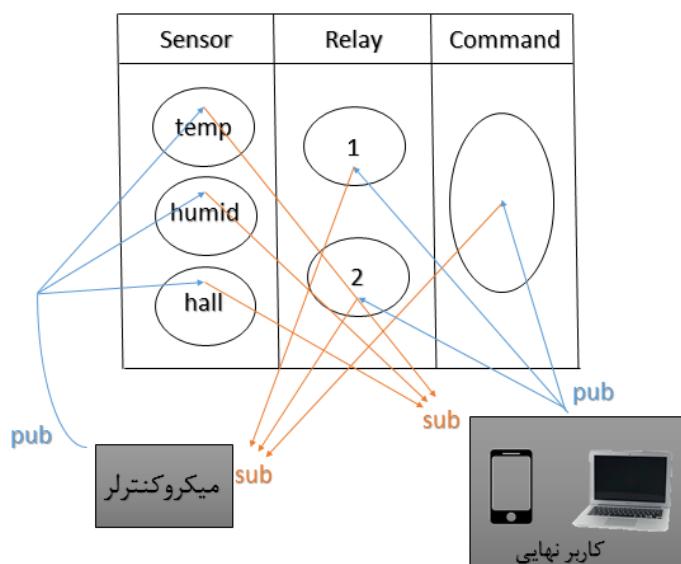
جدول ۱-۳

در ابتدا سه تایپیک برای سنسور ها در نظر می گیریم که یک تایپیک را برای سنسور دما، یک تایپیک را برای سنسور رطوبت و یک تایپیک را نیز برای ارسال توان مصرفی قطعه در نظر می گیریم .

حال می‌بایست کاربر نهایی (مثلا داشبورد) روی این تاپیک مشترک شود تا بتواند اطلاعات ارسالی از میکروکنترلر را دریافت کند.

همچنین یک تاپیک دیگر وجود دارد که برای ارسال درخواست اطلاعات به سنسورها است، به این نحو که میکروکنترلر روی تاپیک command\مشترک می‌شود و کاربر نهایی فرستنده می‌باشد. در صورتی کاربر نهایی مقدار "0" را ارسال کند میکروکنترلر بر اساس پروتکل یاد شده در فصل دوم، میزان دما از DHT22 خوانده و در تاپیک \sensor\temp قرار میدهد. در صورتی که کاربر نهایی مقدار "1" را ارسال کند، میکروکنترلر بر اساس پروتکل یاد شده در فصل دوم، درصد رطوبت را از DHT22 خوانده و به تاپیک \sensor\humid ارسال می‌کند. و در صورتی که کاربر نهایی مقدار "2" را ارسال کند میکروکنترلر بعد از خواندن مبدل آنالوگ به دیجیتال، میزان جریان مصرفی را کالیبره کرده و توان مصرفی را حساب می‌کند. در تاپیک \sensor\hall ارسال می‌کند.

همچنین دو تاپیک دیگر برای کنترل رله‌ها باید ایجاد کرد به این صورت که میکروکنترلر روی دو تاپیک \relay\1 و \relay\2 مشترک شود، به علاوه کاربر نهایی برای روشن یا خاموش کردن مدار عملگر عدد صفر یا یک را برای کنترل عملگر مورد نظر ارسال می‌کند.



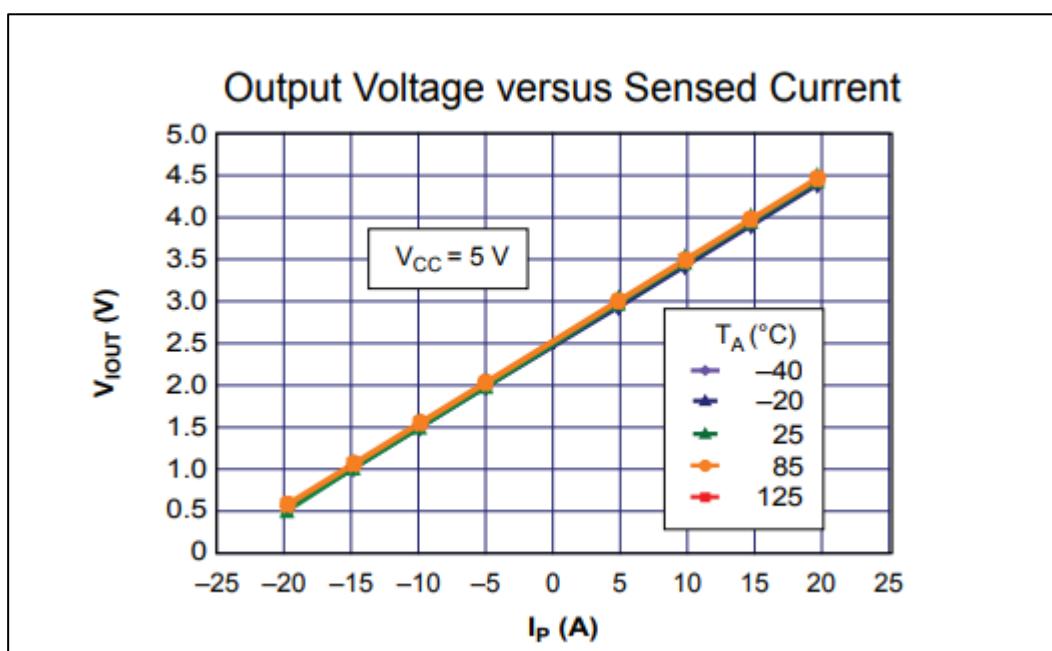
### ۳-۳-۵- بروگ نویسی حسگرها

با توجه به طراحی مفهومی محصول نهایی ، از دو حسگر ، توان ، دما و رطوبت استفاده شد. در این بخش به بررسی نحوه خواندن از آنها و کالیفر کردن آن پرداخته می شود.

### ۳-۳-۵-۱- حسگر توان [7]

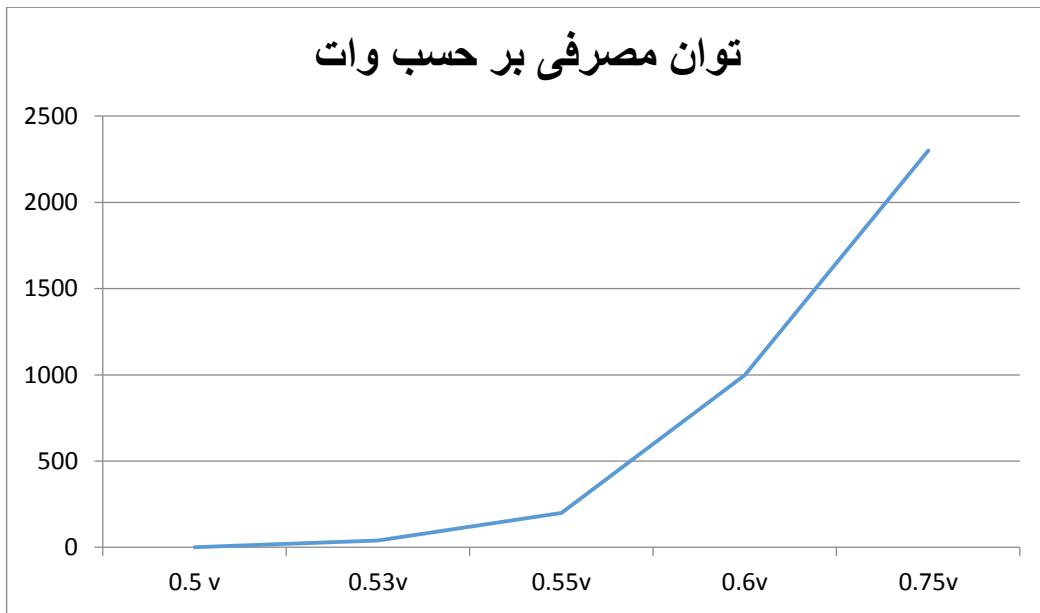
همانطور که در شماتیک مطرح شد ، خروجی این سنسور آنالوگ مطابق شکل ۲۹-۳ می باشد، و برد تغییرات آن بین ۰,۵ ولت تا ۴,۵ ولت می باشد، به دلیل اینکه مبدل آنالوگ به دیجیتال ما تنها در بازه ۰-۱۰ ولت میتواند فعالیت کند، خروجی سنسور را با استفاده از دو مقاومت  $1k, 3.9k$  در بازه ۰-۱۰ ولت منتقل نمودیم و برد بازه مذکور به صورت  $0,1 \text{ تا } 0,91$  تغییر کرد که تابع مذکور را با تقریب دو نقطه و خط به صورت  $I = 40v - 20$  ترتیب داد. که یعنی اگر ولتاژ  $0,91$  باشد جریان عبوری  $20$  آمپر و اگر ولتاژ خروجی  $0,1$  باشد جریان عبوری  $20$  آمپر در جهت معکوس می باشد. و ولتاژ  $0,5$  به معنای عدم عبور جریان است. همچنین برای محاسبه توان ولتاژرا  $220$  ولت گرفته و توان را به صورت  $I = 220(40v - 2)$

در نظر می گیریم.



شکل ۲۹-۳ بخشی از برگه اطلاعات حسگر ها

حال باید فرایند کالیبراسیون را انجام شود و مقادیر نمودار را به صورت تک نقطه بررسی کرد و تابعی جدید تشکیل داد که از نقاط فوق می‌گذرد ، برای دقیق‌تر کار از یک مولتی متر کلمپ استفاده شد تا به صورت دقیق میزان توان را اندازه‌گیری کنیم. و منحنی را تشکیل دهیم منحنی را با فرض ۵ نقطه زیر ، به صورت زیر محاسبه شد:



که برای نوشتن کد کالیبراسیون ، عدد را به بازه های ۵ قسمتی تقسیم کرده و هر کدام از اعداد قرائت شده را در خط مربوط به بازه قرار می دهیم ، الگوریتم مذکور به این صورت است:  
اگر مقدار ولتاژ کوچتر از ۰,۵ باشد : فاصله‌ی عدد را تا نیم حساب کرده و قدر مطلق آن را به ۰,۵ اضافه میکنیم ، بعد بررسی میکنیم، البته این اتفاق در این پروژه رخ نخواهد داد. حال برای ولتاژ های بزرگتر از ۰,۵ خواهیم داشت :

$$P = 1333v - 666 \quad \text{اگر ولتاژ بین ۰,۵ تا ۰,۵۳ باشد}$$

$$P = 8000v - 4200 \quad \text{اگر ولتاژ بین ۰,۵۳ تا ۰,۵۵ باشد}$$

$$P = 16000v - 8600 \quad \text{اگر ولتاژ بین ۰,۵۵ تا ۰,۶ باشد}$$

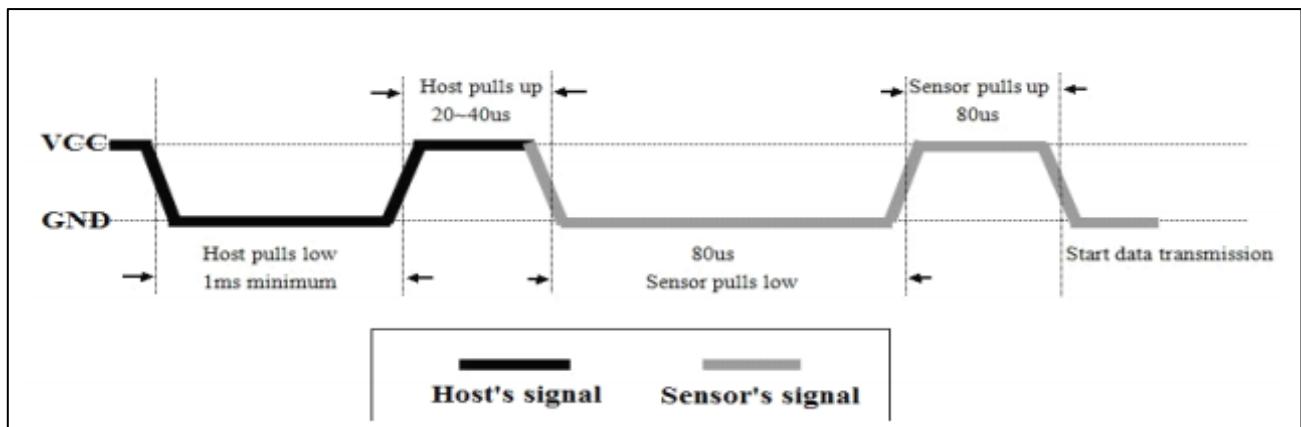
$$P = 86666v - 4200 \quad \text{اگر ولتاژ بین ۰,۶ به بالا باشد}$$

به این شکل سنسور هال یا جریان یا به قولی دیگر توان ، کالیبره شد.

### ۲-۵-۳-۳ حسگر دما و رطوبت [8]

به طور کلی نحوه ارتباط با این سنسور به این صورت است که میکروکنترلر به سمت سنسور ، درخواستی را ارسال می کند ، نحوه ارسال در خواست را در شکل ۳۰-۳ مشاهده می کنیم ، این ارسال به این نحو است که میکروکنترلر پایه پول آپ شده(مشترک) را به مدت حداقل ۱ میلی ثانیه صفر می کند و مجدد به مدت ۲۰ میکرو ثانیه یک می کند و بعد از آن خود را برای خواندن پایه مشترک آماده می کند.

بعد از درخواست میکرو کنترلر نوبت به اعلام آمادگی سنسور برای ارسال اطلاعات می شود. به همین منظور سنسور پایه مشترک را به مدت ۸۰ میکرو ثانیه صفر و بعد از آن به همان میزان یک می کند . (شکل ۳۰-۳) و پس از آن ارسال اطلاعات را شروع می کند.



شکل ۳۰-۳ نحوه درخواست به سنسور 22

نحوه ارسال اطلاعات به این نحو است که، ابتدا ۱۶ بیت داده مربوط به سنسور رطوبت و بعد از آن ۱۶ بیت داده برای سنسور دما و در نهایت ۸ بیت داده برای صحت اطلاعات ارسال شده ، فرستاده می شود.

از ۱۶ بیت داده ارسالی برای سنسور رطوبت ۸ بیت اولیه آن به رقم صحیح درصد رطوبت ، و ۸ بیت ثانویه آن به ارقام بعد ممیز درصد رطوبت اختصاص دارد ، همچنین برای ۱۶ بیت اطلاعات سنسور دما نیز به

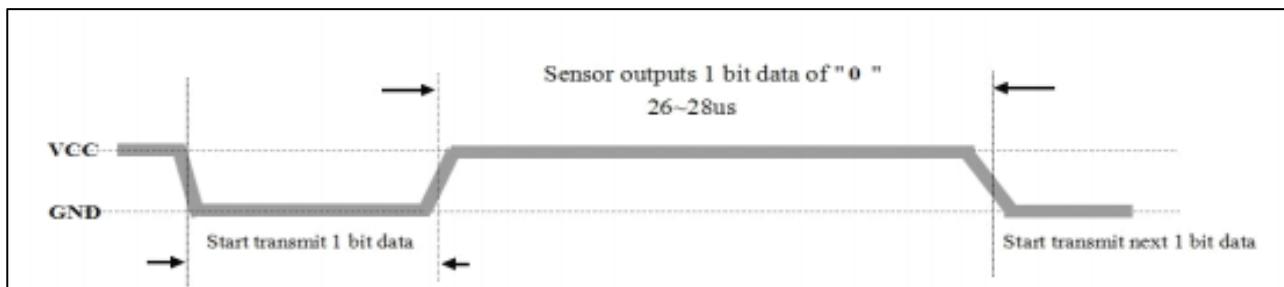
همین نحو است. چرا که این سنسور می‌بایست با پردازنده‌های ۸ بیتی هم به سادگی کار کند و اگر قرار می‌بود ۱۶ بیت داده را به صورت مستقیم ارسال کند، روند پردازشی را برای پردازنده‌ها کمی پیچیده‌تر می‌کرد.

حال به تبیین ۸ بیت، داده‌ی صحیح اطلاعات پرداخته می‌شود، این ۸ بیت حاصل جمع هر ۸ بیت ارسالی است. مثلاً اگر عدد ۱۱۱۰ ۱۱۱۱ ۱۱۱۰ ۰۰۰۱ ۰۱۰۱ ۱۱۱۱ ۱۱۱۰ ۱۰۰۰ ۰۰۰۱ ۰۰۱۰ ۱۱۰۰ ۰۰۰۰ ۰۰۰۱ ارسال شود.

۰۰۰۰ ۰۰۱۰+۱۰۰۰ ۱۱۰۰+۰۰۰۰ ۰۰۰۱+۰۱۰۱ ۱۱۱۱=۱۱۱۰ ۱۱۱۰ اطلاعات ارسال شده اطمینان حاصل نمود. در غیر این صورت می‌توان دوباره درخواست را فرستاد.

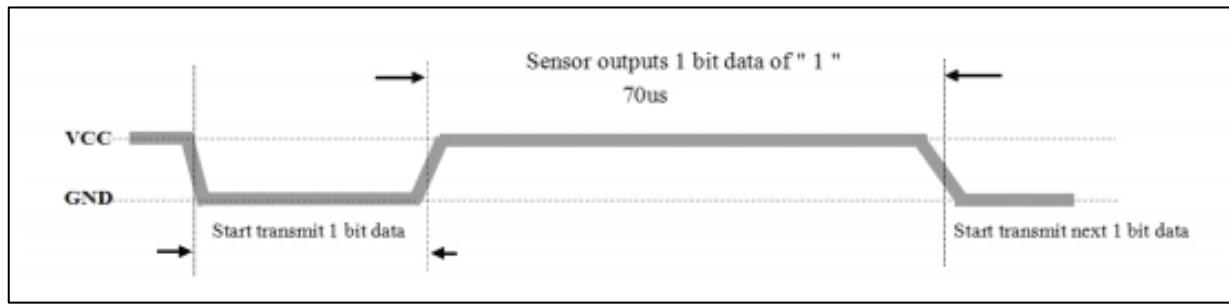
حال نحوه ارسال صفر و یک را بررسی می‌کنیم.

برای ارسال صفر: به این صورت است که ابتدا سنسور، خط را صفر کرده و بعد به مدت ۲۶ تا ۲۸ میکرو ثانیه خط را یک نگه می‌دارد. مطابق شکل ۳۱-۳



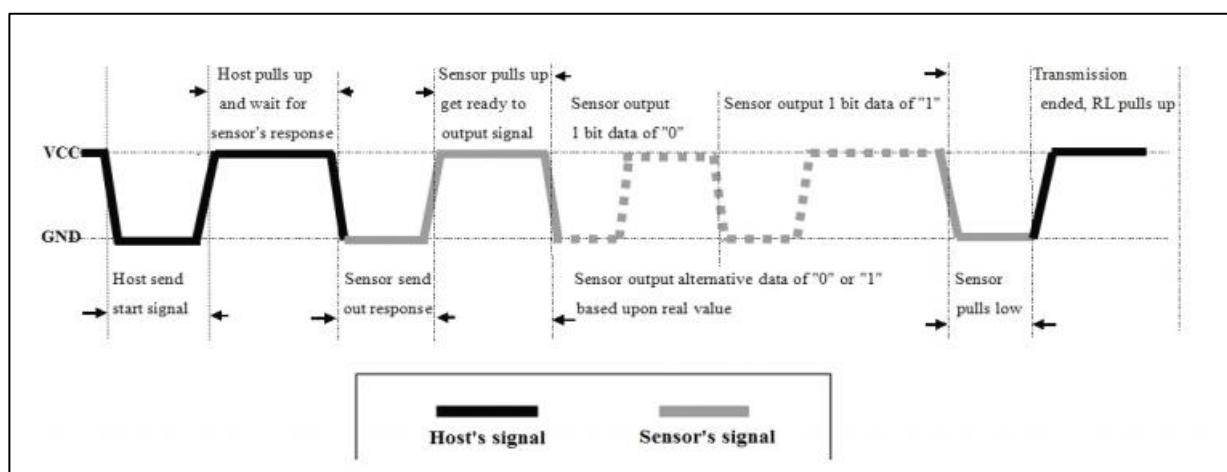
شکل ۳۱-۳ مفهوم ارسال صفر در ارتباط با DHT22

برای ارسال یک: به این صورت است که ابتدا سنسور، خط را صفر کرده و بعد به مدت ۷۰ میکرو ثانیه خط را یک نگه می‌دارد. مطابق شکل ۳۲-۳



شکل ۳۲-۳ مفهوم ارسال یک در ارتباط با DHT22 حال باتوجه روند فوق باید بر روی میکروکنترلر ، وقفه موج بالا رونده را فعال کرده و زمان را شمارش کنیم تا به مفهوم صفر یا یک ارسال شده پی ببریم و پس از ۴۰ بیت دریافتی پایه مذکور را در حالت نوشتمن قرار دهیم.

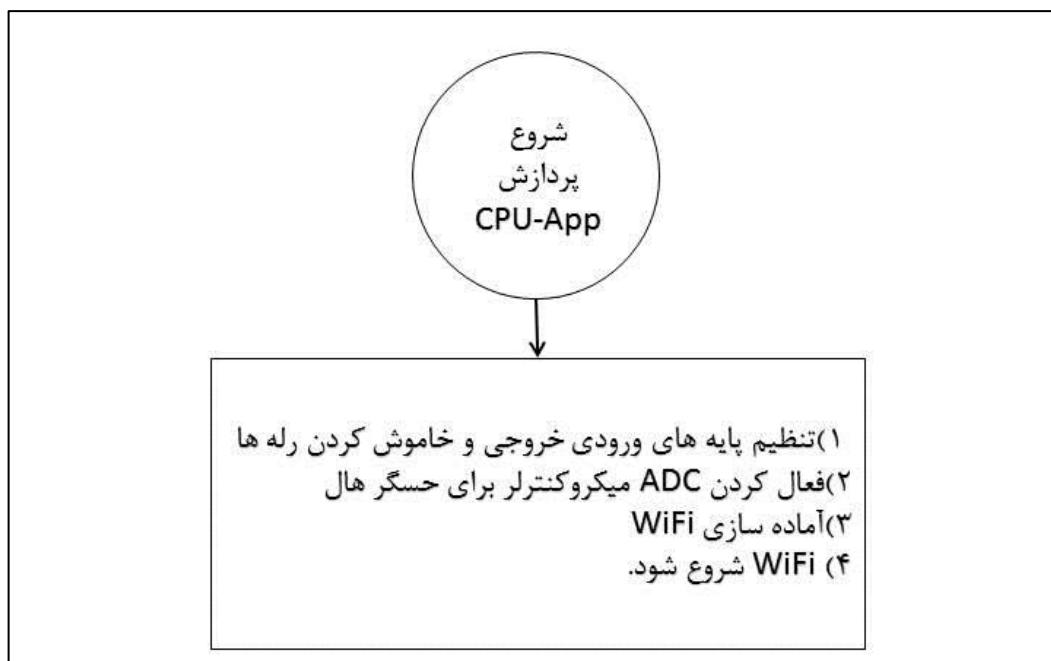
به طور کلی روند زیر (شکل ۳۲-۳) خلاصه ای از خواندن سنسور را در اختیار ما قرار می دهد.



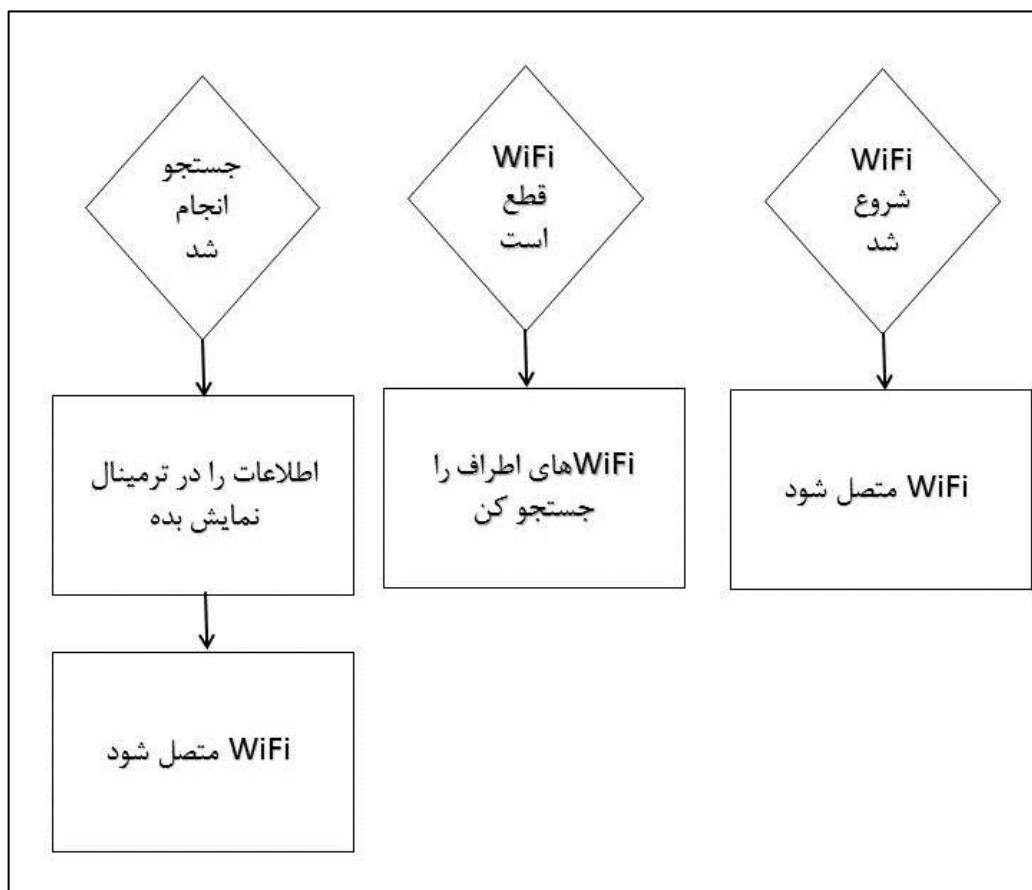
شکل ۳۳-۳ صورت کلی ارتباط با DHT22

### ۳-۳-۶ - برنامه ساخته شده

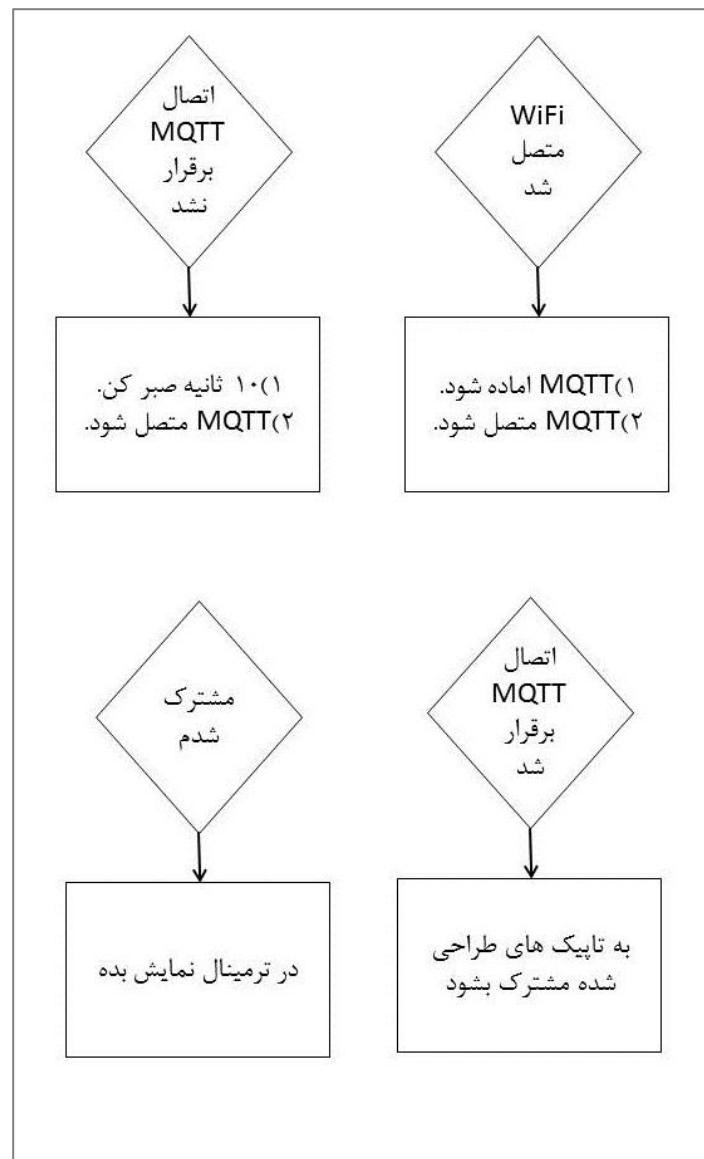
در این قسمت فلوچارت برنامه ساخته شده ، قرار گرفت. پس از توضیحات انجام شده در ۱-۳-۳ ادامه روند اجرای برنامه را فلوچارت ۱-۳ مشاهده می شود. سپس در فلوچارت ۲-۳ نحوه اتصال wifi را بررسی می شود. و در فلوچارت ۳-۳ نحوه آماده سازی و اتصال به سرور MQTT مشخص شده. همچنین روند برنامه و تبادل اطلاعات بین سرور و کلاینت ، دستورات عملگر ها ، هنگام قرائت حسگر ها را در فلوچارت ۴-۳ میتوانیم نظاره کنیم.



فلوچارت ۱-۳



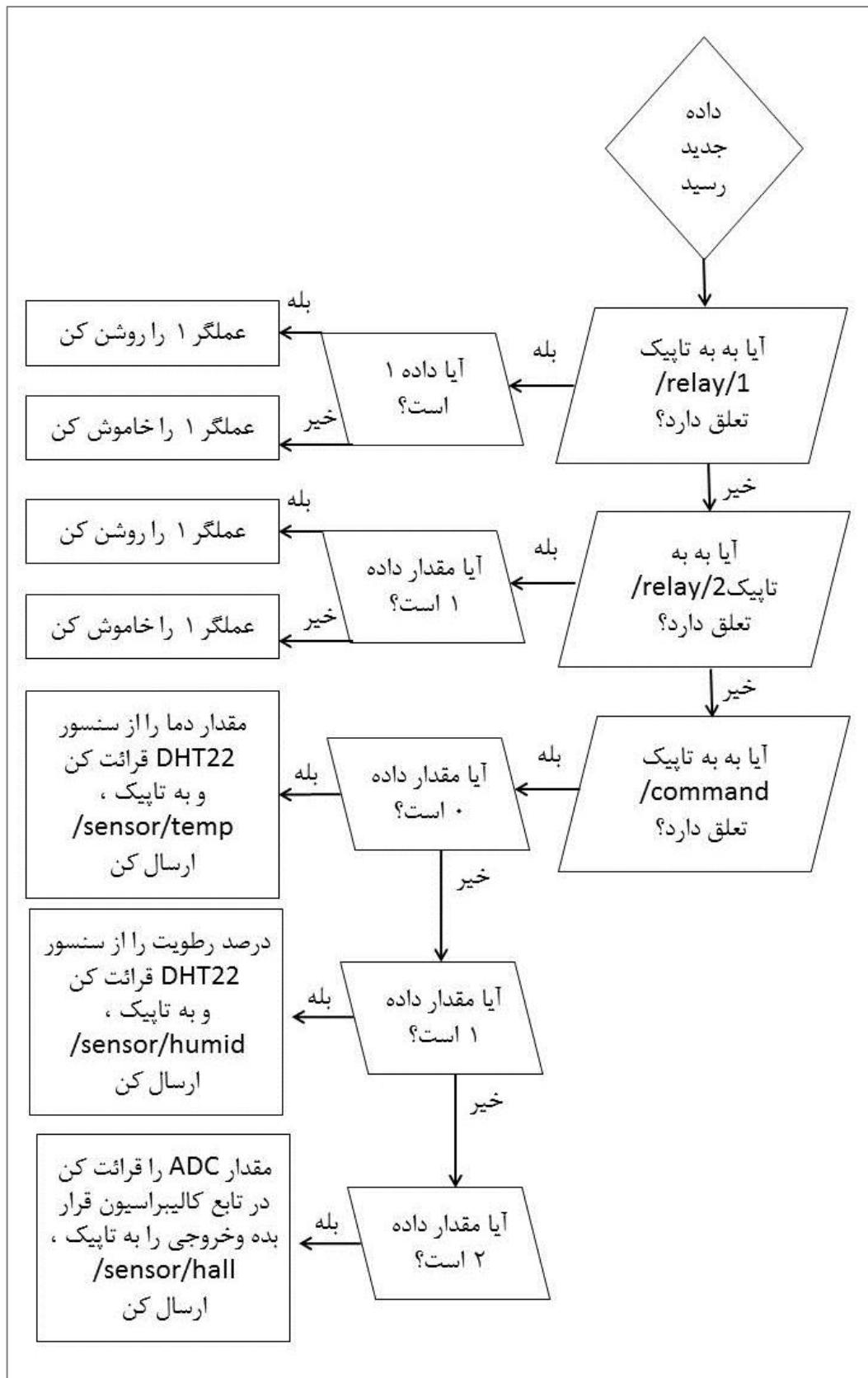
فلوچارت ۲-۳



فلوچارت ۳-۳

تاپیک هایی که پریز هوشمند به عنوان کاربر MQTT می‌باید در آن ها مشترک شود عبارت اند از:

/command    /relay/1    /relay2



۴-۳ فلوچارت

## فصل ۴- نتیجه‌گیری و جمع بندی

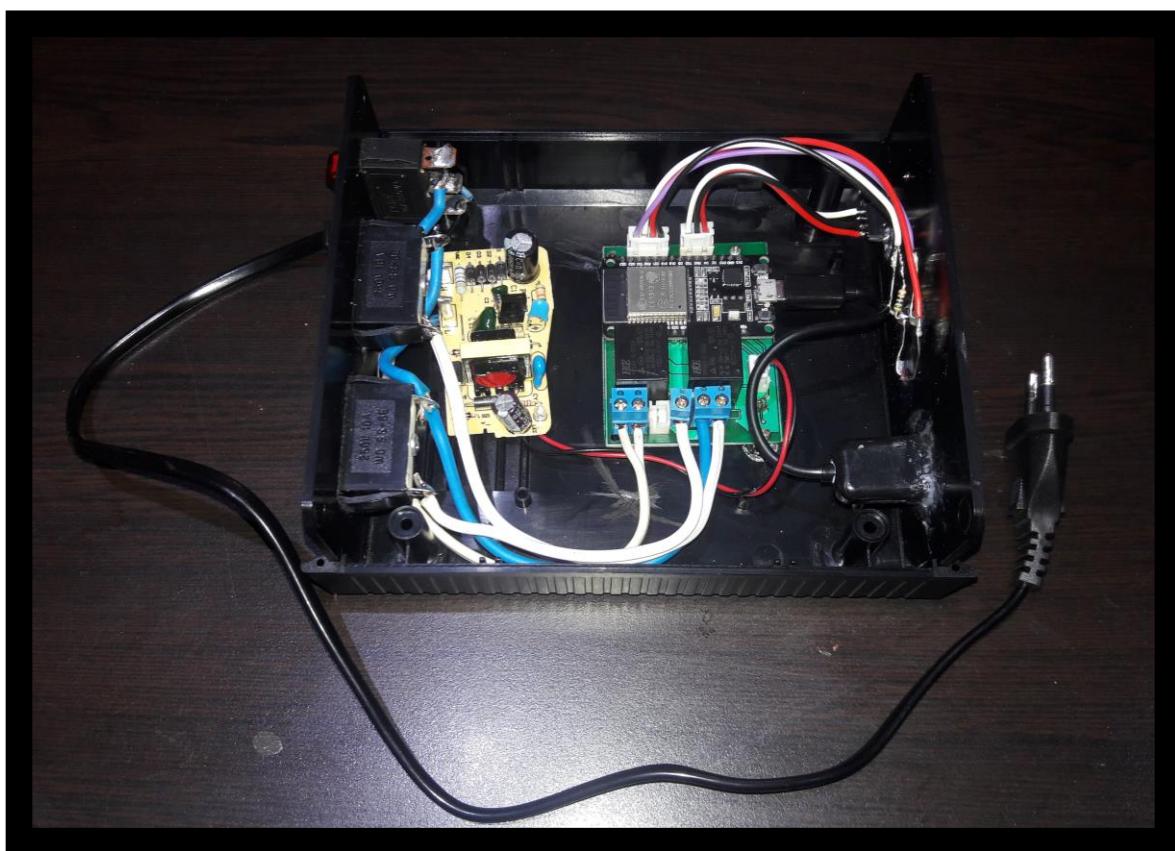
### ۱-۴- روش کار

روش کار با دستگاه فوق بسیار ساده است ، ابتدا دوشاخه پریز هوشمند را در به برق ۲۲۰ ولت متناوب ، متصل می‌کنیم. دستگاه به صورت مستقیم به دنبال یک access point که قبلاً در آن تعریف شده می‌گردد ، تا به آن متصل شود. بعد از برقراری ارتباط به سرور MQTT متصل می‌شود. با برقراری جلسه(Session) چراغ سبز رنگ روی جعبه شروع به چشمک زدن می‌کند که به این است که دستگاه آماده به کار می‌باشد.

حال می‌توانیم از طریق اپلیکشن اندروید به کلود متصل شده و پریز هوشمند خود را کنترل کنیم و از آن لذت ببریم.

### ۲-۴- مثال های تصویری

در این بخش تصاویری از داخل و خارج محصول اولیه و محصول نهایی را در اختیار خواننده قرار می‌گیرد: تصاویر مدار شماره ۱ تا ۶ متعلق به محصول نهایی و تصاویر ۷ و ۸ مربوط به محصول اولیه می‌باشد.



تصویر مدار ۱



تصویر مدار ۲



تصویر مدار ۳



تصویر مدار ۴



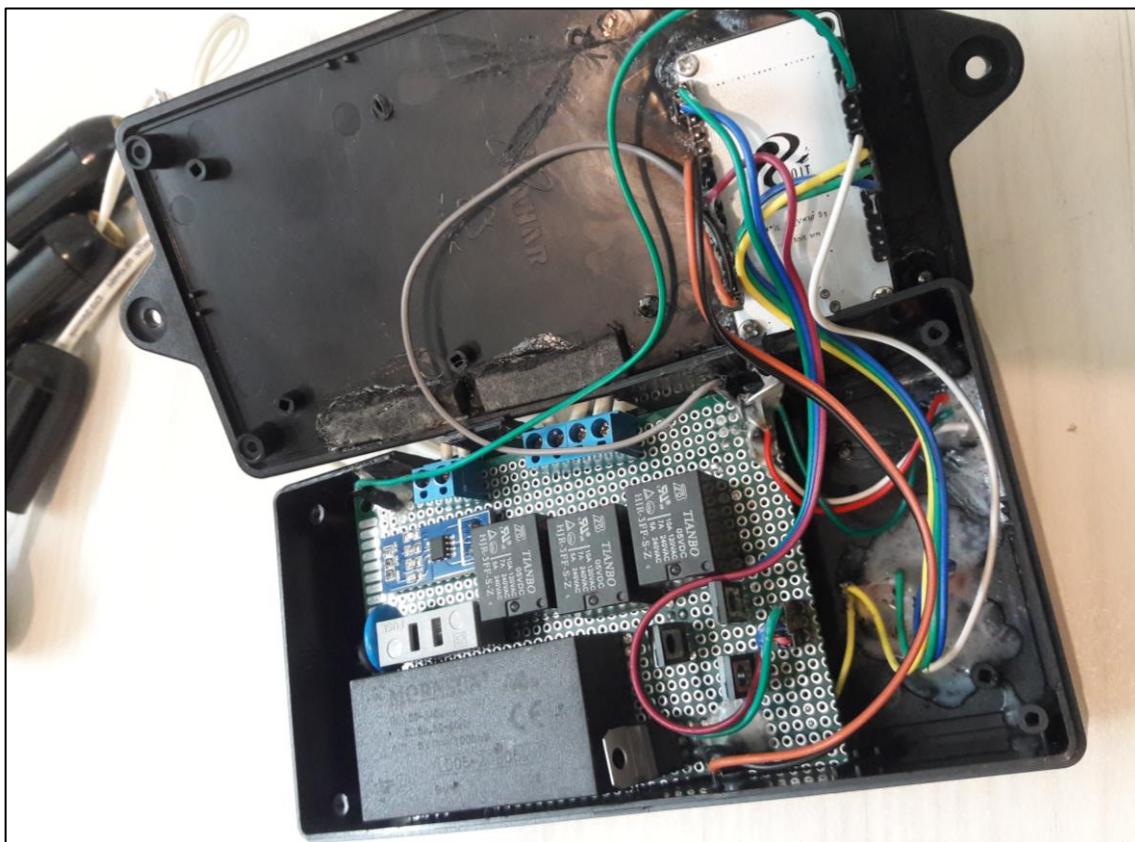
تصویر مدار ۵



تصویر مدار ۶



تصویر مدار ٧



تصویر مدار ٨

## ۳-۴ اطلاعات عملکردی

همانطور که از جدول زیر مشاهده می‌کنیم هنگامی که مودم دستگاه خاموش است، به میزان ۲۰ تا ۲۵ میلی آمپر جریان استفاده می‌کند. و برای wifi در هنگام ارسال پیام ۱۸۰ میلی آمپر و هنگام فعال بودن ۱۰۰ میلی آمپر جریان می‌کشد. پس به طور میانگین میکرو کنترلر ما ۱۳۰ میلی آمپر جریان می‌کشد، با توجه به ولتاژ ۳,۳ ولت میکروکنترلر، توانی برابر ۴,۰ وات مصرف می‌کند.

Power mode	Description			Power consumption	
Active (RF working)	Wi-Fi Tx packet			Please refer to Table 14 for details.	
	Wi-Fi/BT Tx packet				
	Wi-Fi/BT Rx and listening				
Modem-sleep	The CPU is powered on.	240 MHz *	Dual-core chip(s)	30 mA ~ 68 mA	
			Single-core chip(s)	N/A	
		160 MHz *	Dual-core chip(s)	27 mA ~ 44 mA	
			Single-core chip(s)	27 mA ~ 34 mA	
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA	
			Single-core chip(s)	20 mA ~ 25 mA	
Light-sleep	-			0.8 mA	
Deep-sleep	The ULP co-processor is powered on.			150 µA	
	ULP sensor-monitored pattern			100 µA @1% duty	
	RTC timer + RTC memory			10 µA	
Hibernation	RTC timer only			5 µA	
Power off	CHIP_PU is set to low level, the chip is powered off.			0.1 µA	

Table 14: RF Power-Consumption Specifications

Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11b, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11g, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

همچنین، هنگامی که رله روشن است، از مدار کنترل کننده رله جریان ۱۰۰ میلی آمپر کشیده می‌شود و با توجه به منبع تغذیه ۵ ولتی توان ۰,۵ وات نیز صرف روشن نگه داشته شدن رله می‌شود. پس به طور کلی

توان مصرفی دستگاه در حداکثر حالت ۱,۵ وات می‌باشد. همچنین هر کدام از رله‌ها توانایی عبور ۱۰ آمپر را از خود دارند و حسگر هال میتواند تا جریان ۲۰ آمپر را اندازه گیری کند.

با توجه به اطلاعات فوق جدول اطلاعات عملکردی به نحو زیر است:

مشخصات فیزیکی	
ابعاد	17*14*5.5cm
وزن	400 gr
تعداد پریز	2

جدول ۱-۴ مشخصات فیزیکی

پرازنده	ESP32
دروازه ارتباطی	WiFi (802.11 b,g)
لایه انتقال	TCP/IP
لایه اپلیکیشن	MQTT
ادرس سرور	mqtt://m12.cloudmqtt.com:12786
User/pass	snmopxmt/5KI-PQXITuv6

جدول ۲-۴ مشخصات ابری

مشخصات فنی	Min	---	Max	
توان مصرفی دستگاه	0.5		1.5	W
توان خروجی			2200	W
ولتاژ خروجی		220		V
جريان خروجی			10	A
ولتاژ ورودی		220		V
دماي کاري مناسب	10		45	C°

جدول ۳-۴ مشخصات فنی

جدول ۱-۴ مربوط به اطلاعات فیزیکی محصول نهایی است. که در آن به ابعاد، وزن و تعداد پریزهای به کار رفته در محصول اشاره شده است.

جدول ۲-۴ به شیوه‌ی ارتباطی میکروکنترلر مربوط می‌باشد.

مشخصات فنی محصول نهایی را در جدول ۳-۴ مشاهده می‌کنیم.

## فصل ۵ - پیشنهادات

در این فصل به پیشنهادات و فرایندهای توسعه ای که دقیقاً روی همین سخت افزار می‌توان انجام داد و در صورت پیاده سازی، تماماً ارزش افزوده مطرح می‌شود، پرداخته می‌شود.

### ۱-۵ پیشنهاد شماره ۱

هنگامی که میکروکنترلر هنوز به access point متصل نشده است، میتوان بخش wifi خود را در مود<sup>۱</sup> AP قرار داد و از طریق http و پروتکل tcp یک صفحه را روی مرورگر به کاربر نشان داد. که در آن صفحه می‌توان اطلاعات access point های محیط که پیشتر در wifi scan و ترمینال نمایش داده شد را نمایش داد. و از طریق همین پروتکل ssid و pass ، مودم مورد نظر را دریافت کرد. و بعد از این که محصول ip گرفت ، میکروکنترلر ، مود AP خود را قطع کند تا انرژی کمتری مصرف شود.

### ۲-۵ پیشنهاد شماره ۲

هنگامی که بخواهیم محیطی را هوشمند سازی کنیم ، تعداد قطعات هوشمند بالا میرود در چنین سناریویی ، همه‌ی دستگاه‌ها نمی‌توانند به مودم متصل شوند پس می‌بایست تدبیر دیگری اندیشید. برای رفع این مشکل ، از آن جایی که ESP32 از BLE پشتیبانی می‌کند. می‌توان تنها از یک ESP32 استفاده کرد تا به access point متصل شود و برای سایر محیط از مژول‌های بلوتوث استفاده کرد و شبکه‌ای با دروازه<sup>۳</sup> بلوتوث ایجاد کرد که با پروتکل COAP با یکدیگر در ارتباط هستند. این کار علاوه بر حل مشکل فوق ، دو مزیت برای ما دارد. ۱) کاهش هزینه ساخت محصول ۲) کاهش توان مصرفی

<sup>1</sup> access point

<sup>2</sup> بلوتوث کم مصرف

<sup>3</sup> Gate way

### -۳-۵ پیشنهاد شماره ۳

با توجه به بررسی روند پروژه ، یکی از مشکلات پیش رو مشکل پلتفرم و کلود بود که نبود یک پلتفرم و کلود MQTT قوی برای کشورمان احساس می شد. به عنوان پیشنهاد سوم ، طراحی یک کلود MQTT با قابلیت های بالا (نظیر دیتابیس قوی ، امنیت بالا و...) در درجات بالایی قرار دارد.

### -۴-۵ پیشنهاد شماره ۴

برای مراحل بعدی این پروژه ، میتوان به ارائه اپلیکشن موبایل(اندروید و iOS) و کامپیوتر منحصر به فرد ، اشاره کرد. که جنبه های ارائه پروژه را بهبود می دهد. چرا که محدودیتی برای نوع وسیله متصل موجود نمی باشد و هر وسیله ای که بتواند به اینترنت متصل شود می تواند با این دستگاه کار کند.

### -۵-۵ پیشنهاد شماره ۵

تغییر کاربری پروژه برای بازار های هدف بزرگتر است . مثلا ؛ مسائل نظارت و امنیت ، مسائل کشاورزی ، مسائل صنعتی ، مسائل حوزه سلامت و بهداشت و....

### -۶-۵ پیشنهاد شماره ۶

بهبود طراحی جعبه محصول برای کاربرد های مختلف.

### -۷-۵ پیشنهاد شماره ۷

تغییر دروازه از Wifi به ارتباطات برد بلند مثل NBLoT و Sigfox و LoRaWAN و استفاده از ESP32 که از طریق access point Wifi به میکروکنترلر دیگر که بتوانیم از دروازه های فوق استفاده کنیم، که برد بالاتری دارند و در مواردی کم مصرف تر و با توجه به نوع کاربرد پهنای باند کمتری اشغال می کنند، بهره ببریم.

## فصل ۶- فهرست مراجع

- [1] S. C. Mukhopadhyay, Internet of Things Challenges and Opportunities, Massey University, New Zealand: Springer.
- [2] E. Pietrosemoli, "Wireless standards for IoT," in *International Centre for Theoretical Physics*, Italy.
- [3] "xiaomishop," xiaomi, [Online]. Available: <https://www.xiaomishop.ir/product/xiaomi-smart-socket-plug-2/>.
- [4] ESP32 Series Version 2.5, Espressif Systems, 2018.
- [5] E. Ramsden, Hall-Effect Sensors, Newnes, 2006.
- [6] D. Gilliam, Introduction to Mechatronics, Temperature Sensors, 2003.
- [7] ACS712, Worcester, Massachusetts, U.S.A: Allegro MicroSystems, Inc.
- [8] I. ada, DHTxx Sensors, Adafruit Industries.
- [9] General Notes About ESP-IDF Programming, API Guide, Espressif Systems, 2018.
- [10] E. International Business Machines Corporation (IBM), MQ Telemetry Transport (MQTT) V3.1 Protocol, MQTT.org, 2010.





## پیوست ۱ هرآن چیزی که برای شروع مورد نیاز است

۱) یک رایانه که به سیستم عاملی مانند ، ویندوز و لینوکس یا مک مجهز باشد.

۲) یک Toolchain برای ساختن <sup>۱</sup>یک کاربرد <sup>۲</sup>برای ESP32

۳) یک ویرایشگر متن برای نوشتن برنامه

۴) ESP32 و یک سیم برای اتصال به رایانه

Toolchain چیست؟

در روند ایجاد نرمافزار معمولاً از تعدادی ابزار خاص استفاده می‌شود. فرضا برای نوشتن کد متن یا سورس برنامه (source code) و ذخیره آن در فایل از یک ویرایشگر متن استفاده می‌شود. فایل مزبور باید با کامپایلر کامپایل شود و به همین صورت خروجی هر ابزاری ورودی ابزار بعدی خواهد بود.

زنجیره ابزار (یا چنانکه بیشتر متعارف است ToolChain) در دانش نرمافزار عبارت است از مجموعه‌ای از ابزارها (عمدتاً نرمافزاری) که جهت ایجاد محصولی خاص (که آن هم معمولاً نرمافزار است) باید به ترتیب خاصی به کار روند. چنین زنجیره‌ای ممکن شامل اجزای زیر باشد:

ویرایشگر متن – جهت وارد کردن کد متن

کامپایلر – برای تبدیل کد متن به کتابخانه شی

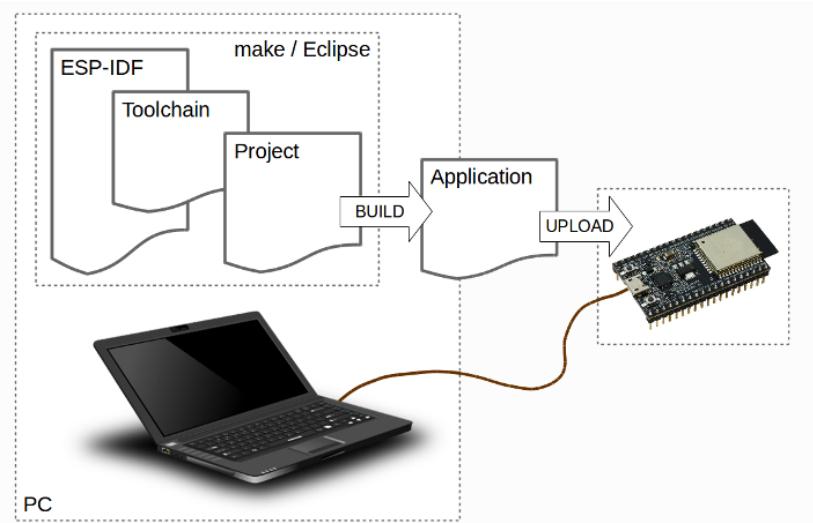
لینکر – ترکیب کتابخانه حاصله با استاندارد کامپایلر و تولید کد قابل اجرا

<sup>1</sup> build

<sup>2</sup> Application

محیط‌های برنامه‌سازی بصری جدید عمدتاً زنجیره ابزار را طوری درون خود پیاده سازی کرده‌اند که برنامه نویس نیازی به درگیر شدن با جزئیات مربوط به آن را ندارد.

#### خلاصه



ESP-IDF مجموعه‌ای از کتابخانه‌ها (components) است.

که از طریق یک Toolchain برنامه‌هایی که توسط کاربر نوشته شده است را به یک برنامه کاربردی تبدیل می‌کند. در پروژه انجام شده از msys32 به عنوان eslips و از toolchain به عنوان محیطی برای نوشتن برنامه که با esp-idf و toolchain هماهنگ است و قابلیت اتصال را دارد، استفاده شد. همچنین در همین محیط فرایند ارسال داده به روی میکرو نیز انجام شد.

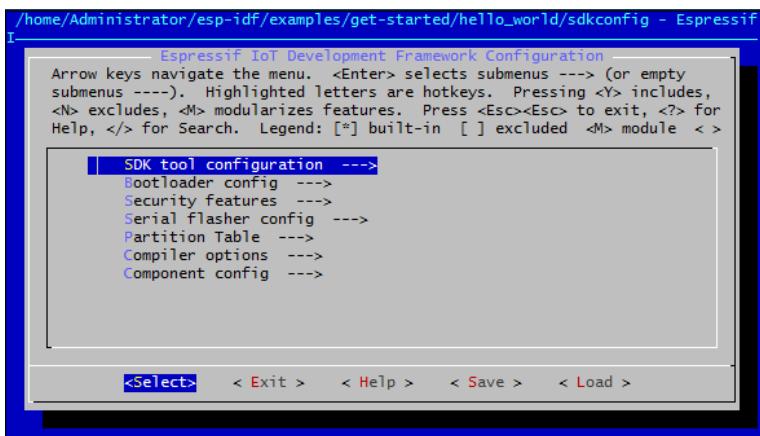
## پیوست ۲ جریان ایجاد برنامه

پیش‌تر توضیح داده شد که ESP-IDF از تعدادی کتابخانه ساخته شده که توسط toolchain یک برنامه تبدیل می‌شود، حال برای تبیین دقیق موضوع میباشد ابتدا چند مفهوم را توضیح دهیم.

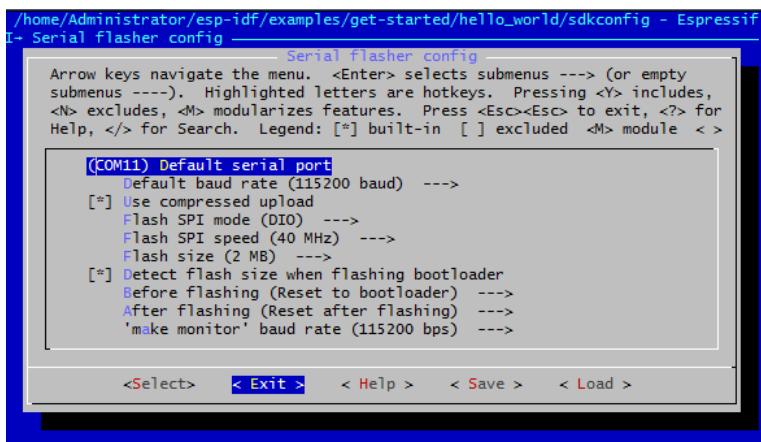
- یک پروژه مسیری است که شامل همه فایل‌ها و تنظیمات، مورد نیاز ساخت یک برنامه را در برداشته باشد، مانند فایل تنظیمات پروژه، داده‌های مربوط به بوت شدن میکرو، داده‌های نوشته شده روی پردازنده‌ها و جدول بخش بندی حافظه و فایل‌هایی که قرار است در سیستم ذخیره شوند و....

- فایل تنظیمات پروژه، فایلی است که در آن میتوان تنظیمات کلی پروژه را در محیطی مانند بوت ذخیره نمود. برای اجرای پروژه، وجود این فایل ضروری است. محیط ذکر شده در شکل‌های زیر

مشخص است:



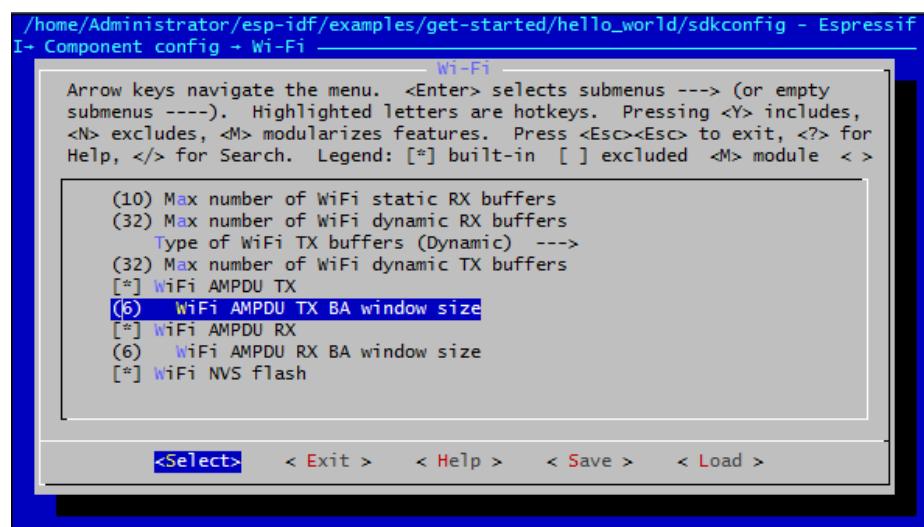
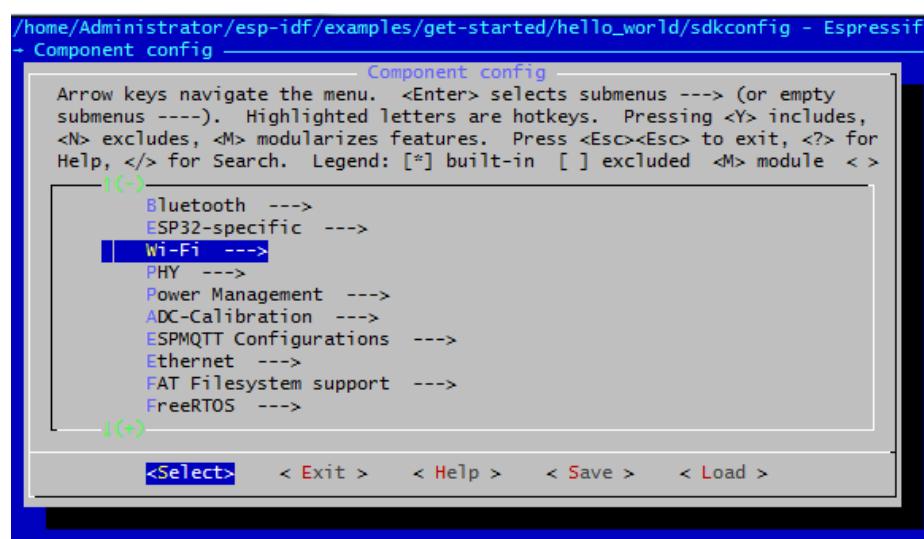
برای مثال پورت و بادریت برای آپلود روی میکرو در این قسمت باید مشخص شود:



یا برای مثالی دیگر میزان تقویت و تضعیف یا... سیگنال وای فای می بایست در قسمت تنظیم شود

به طور کلی هر تنظیمات که مربوط به بخش نرم افزاری پروژه نباشد ، در این قسمت قرار می گیرد.

لازم به ذکر است که تنها در این FirmWare به این سطح از تغییرات دسترسی داریم:



قطعه های مازول واری هستند که از کدهای مستقل کامپایل شده ساخته شده  
اند و به شمایل یک کتابخانه در اختیار کاربر قرار داده می‌شوند. Esp-idf از تعدادی از این قطعه ها  
تشکیل شده که کد ما را به وسیله‌ی toolchain با استفاده از این کامپونت‌ها کامپایل می‌کند.  
این به این معنی است که ESP-idf یخشی از پروژه‌ما نیست، بلکه پروژه‌ما به وسیله‌ی آن ساخته  
می‌شود.

برای مثال یک پروژه از اجزای زیر تشکیل شده است :

```
- myProject/
  - Makefile
  - sdkconfig
  - components/
    - component1/
      - component.mk
      - Kconfig
      - src1.c
    - component2/
      - component.mk
      - Kconfig
      - src1.c
      - include/
        - component2.h
  - main/
    - src1.c
    - src2.c
    - component.mk
  - build/
```

## پیوست ۳ مقایسه محیط هایی برای توسعه ESP32

به عنوان میکروکنترلی ۲ هسته ای که از بلوتوث و وای فای پشتیبانی می کند. در سطح جهانی بسیار محبوب شده تا جایی که این کمپانی در انتهای سال ۲۰۱۷ اعلام کرد که ۱۰۰ میلیون از این قطعه تا کنون تولید شده. پس شرکت های زیادی سرتاسر دنیا برای توسعه این برد ها اقدام کرده اند که در اینجا به خلاصه از آن ها که توسط خود espessif مورد قبول هستند می پردازیم.

### AT Commands

AT commands دستوراتی هستند که از طریق ارتباط سریال به میکرو فرستاده می شوند و همانطور که از نام آن ها پیدا است ، به صورت دستور هستند ، که به این معنا است در صورتی که می خواهیم از این firmware استفاده کنیم باید از یک میکروکنترلر دیگر استفاده کنیم . همانطور که مشخص است دستورات AT راهکار مناسبی برای انجام این پروژه نیست ، چرا که خود ESP32 یک میکروکنترلر کامل است . ولی این دستورات اولین و بهترین گزینه برای آشنایی با امکانات جانی این میکرو هستند. تعدادی از دستورات مربوط به ماژول وای فای و بلوتوث آن را مشاهده می کنیم:

#### BLE-Related AT Commands

AT+BLEINIT—BLE Initialization	AT+CIPSTATUS—Gets the Connection Status
AT+BLEADDR—Sets BLE Device's Address	AT+CIPDOMAIN—DNS Function
AT+BLENANE—Sets BLE Device's Name	AT+CIPDNS—Sets User-defined DNS Servers; Configuration Saved in the Flash
AT+BLESCANPARAM—Sets Parameters of BLE Scanning	AT+CIPSTAMAC—Sets the MAC Address of the ESP32 Station
AT+BLESCAN—Enables BLE Scanning	AT+CIPAPMAC—Sets the MAC Address of the ESP32 SoftAP
AT+BLESCANRSPDATA—Sets BLE Scan Response	AT+CIPSTA—Sets the IP Address of the ESP32 Station
AT+BLEADVPARAM—Sets Parameters of Advertising.	AT+CIPAP—Sets the IP Address of the ESP32 SoftAP
AT+BLEADVDATA—Sets Advertising Data	AT+CIPSTART—Establishes TCP Connection, UDP Transmission or SSL Connection
AT+BLEADVSTART—Starts Advertising	AT+CIPSSLCCONF—Set Configuration of SSL Client
AT+BLEADVSTOP—Stops Advertising	AT+CIPSEND—Sends Data
	AT+CIPSENDEX—Sends Data

#### TCP/IP-Related AT Commands

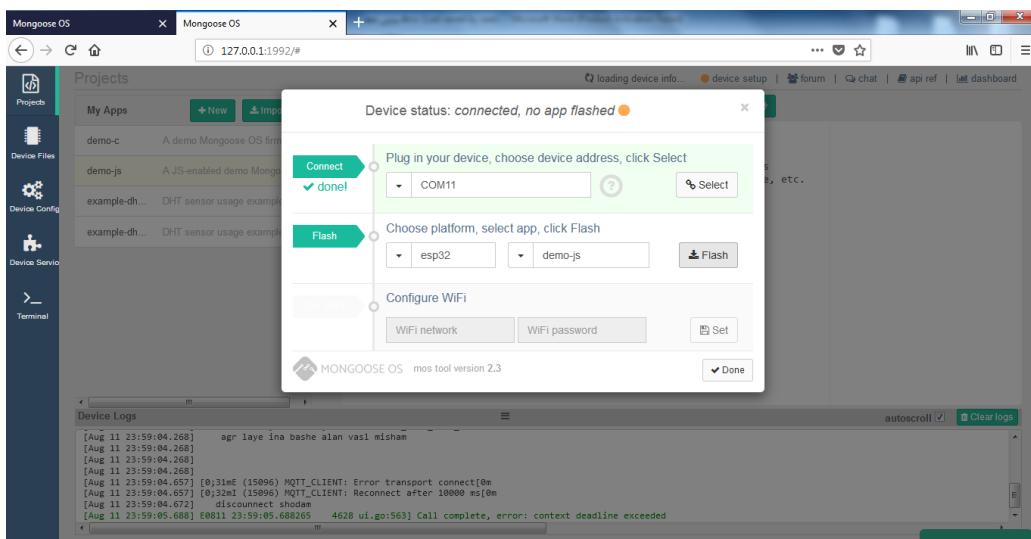
AT+CIPSTATUS—Gets the Connection Status
AT+CIPDOMAIN—DNS Function
AT+CIPDNS—Sets User-defined DNS Servers; Configuration Saved in the Flash
AT+CIPSTAMAC—Sets the MAC Address of the ESP32 Station
AT+CIPAPMAC—Sets the MAC Address of the ESP32 SoftAP
AT+CIPSTA—Sets the IP Address of the ESP32 Station
AT+CIPAP—Sets the IP Address of the ESP32 SoftAP
AT+CIPSTART—Establishes TCP Connection, UDP Transmission or SSL Connection
AT+CIPSSLCCONF—Set Configuration of SSL Client
AT+CIPSEND—Sends Data
AT+CIPSENDEX—Sends Data

## Mongoose OS

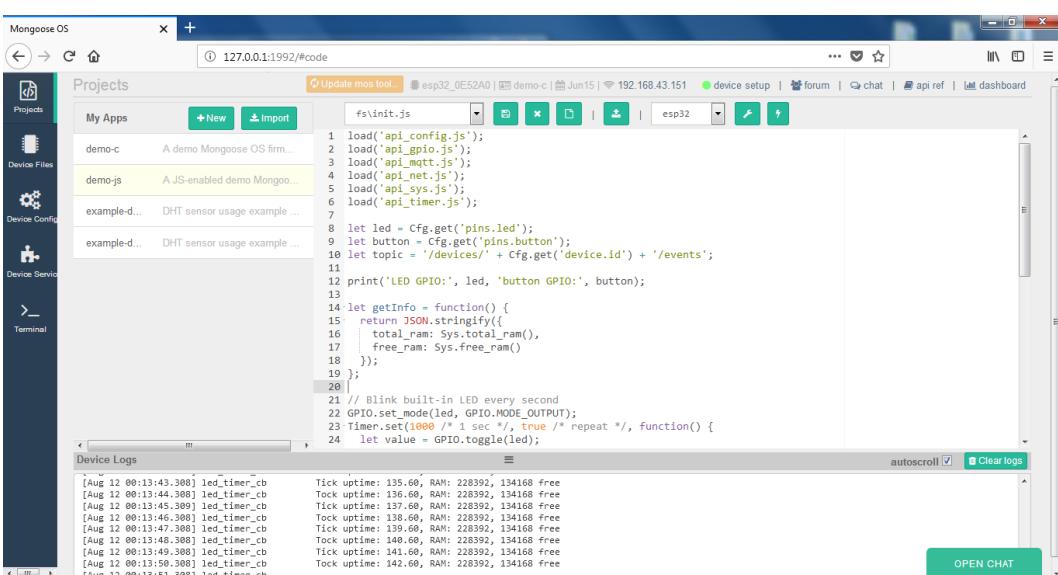
یک سیستم عامل متن باز برای اینترنت اشیاء است که از Mongoose OS پشتیبانی میکند. این سیستم عامل زمان و هزینه ساخت Firmware را تا ۹۰ درصد کاهش می‌دهد. بسیار ساده و قابل اعتماد است و در آن می‌توان با زبان‌های C یا JavaScript شروع به کد زدن کرد.

این محیط همانطور که یاد شد، به خاطر تنظیم کردن ساده میکرو و وجود مثال‌های بسیار کار کردن را بسیار ساده می‌کند، که برای آشنا شدن با امکانات کارها و فرایند‌هایی که می‌توان با میکروکنترلر انجام داد؛ مانند... آشنا شد.

به طور مثال در شکل زیر نحوه اتصال و نصب Firmware را مشاهده می‌کنیم:



و در شکل زیر محیط کد زدن، لگ و محیط کلی صفحه را مشاهده می‌کنیم:



ولی در کنار همه‌ی مزیت سادگی بزرگترین عیب آن ، این است که:

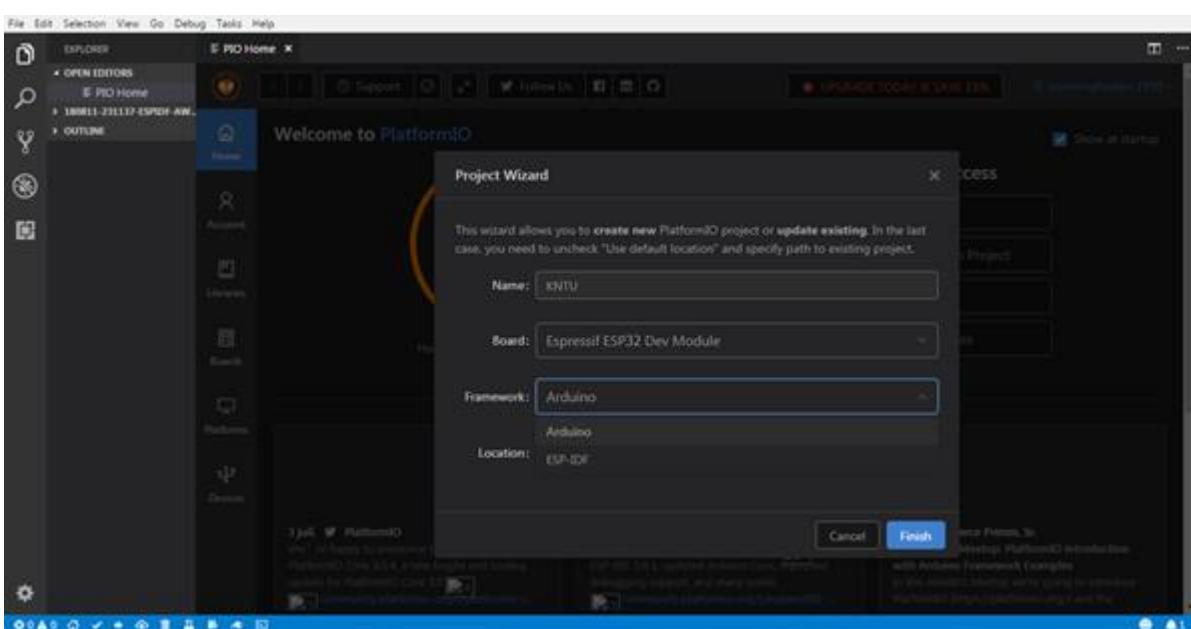
- به خیلی از امکانات ESP32 دسترسی نداریم.
- همچنین سایت اصلی این سیستم عامل برای ایرانیان محدود شده ، و در صورت تمایل به استفاده از این سیستم عامل می‌باشد، از طریق IP سایر کشورها به این سایت متصل شد.

## Platform IO

یک اکوسیستم اپن سورس برای سیستم‌های اینترنت اشیاء است که در آن قابلیت کد زدن ، دیباگ کردن ، تست کردن بخش‌ها و کار کردن با انواع مختلف firmware هاست که از ESP32 نیز پشتیبانی می‌کند.

محیط نوشتن کد برای این اکوسیستم Microsoft VS Code می‌باشد، همچنین firmware شده در آن برای ESP32 همان ESP-IDF و Arduino می‌باشد. که با زبان C,C++ برنامه‌ها نوشته می‌شوند.

تصاویری از محیط کاری این اکوسیستم مشاهده می‌کنیم:



```
C subscribe_publish_sample.c
307     wifi_config_t wifi_config = {
308         .sta = {
309             .ssid = EXAMPLE_WIFI_SSID,
310             .password = EXAMPLE_WIFI_PASS,
311         },
312     };
313     ESP_LOGI(TAG, "Setting WiFi configuration SSID %s...", wifi_config.sta.ssid);
314     ESP_ERROR_CHECK( esp_wifi_set_mode(WIFI_MODE_STA) );
315     ESP_ERROR_CHECK( esp_wifi_set_config(WIFI_IF_STA, &wifi_config) );
316     ESP_ERROR_CHECK( esp_wifi_start() );
317 }
318
319 void app_main()
320 {
321     // Initialize NVS.
322     esp_err_t err = nvs_flash_init();
323     if (err == ESP_ERR_NVS_NO_FREE_PAGES) {
324         ESP_ERROR_CHECK(nvs_flash_erase());
325         err = nvs_flash_init();
326     }
327     ESP_ERROR_CHECK( err );
328
329     initialise_wifi();
330 #ifdef CONFIGMBEDTLS_DEBUG
331     const size_t stack_size = 36*1024;
332 #else
333     const size_t stack_size = 36*1024;
334 #endif
335     xTaskCreatePinnedToCore(&aws_iot_task, "aws_iot_task", stack_size, NULL, 5, NULL, 1);
336 }
337 }
```

از مزایای آن می‌توان به در دسترس بودن همه‌ی بخش‌ها اشاره کرد و به سادگی می‌توان به هر کتابخانه‌ای دسترسی پیدا کرد چرا که visual studio به سادگی با متصل شدن به سرور‌ها هر آن چیزی که شما نیاز دارید را دانلود می‌کند، همچنین خیلی از بخش‌هایی که در طراحی فعلی پروژه انجام شد، در صورت استفاده از VS دیده نمی‌شود مانند .toolchain

Zerynth Studio

Zerynth روی ESP32 برنامه نویسی کنند. Python & hybrid به برنامه نویسان حوزه اینترنت اشیاء این اجازه را می‌دهد تا با زبان محیط از این نرم افزار را در تصویر زیر مشاهده می‌شود:

The screenshot shows the Zerynth Studio interface with the following details:

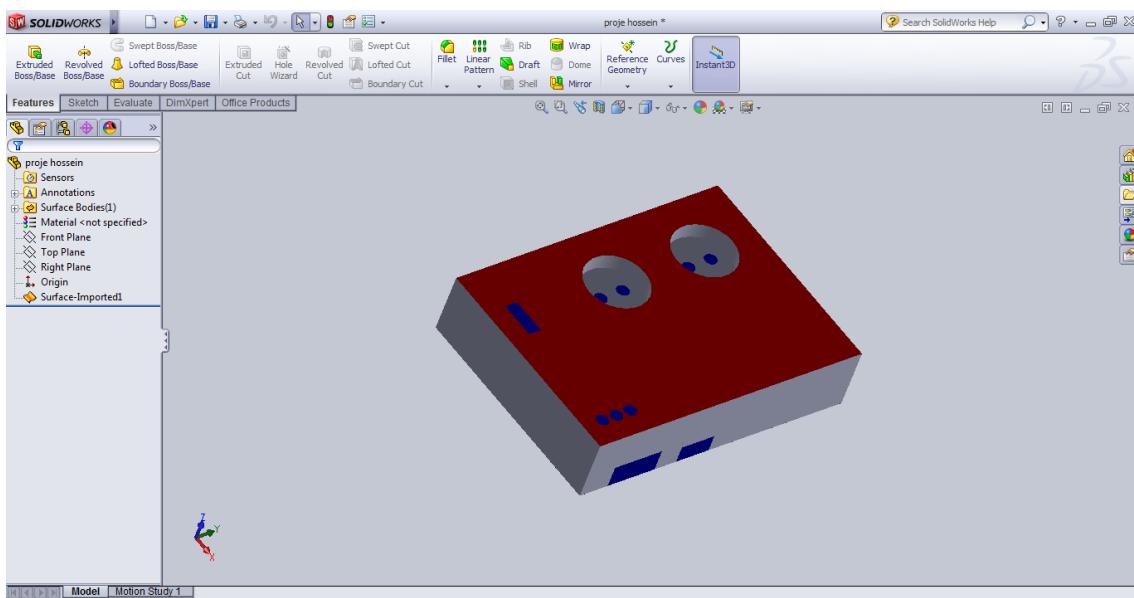
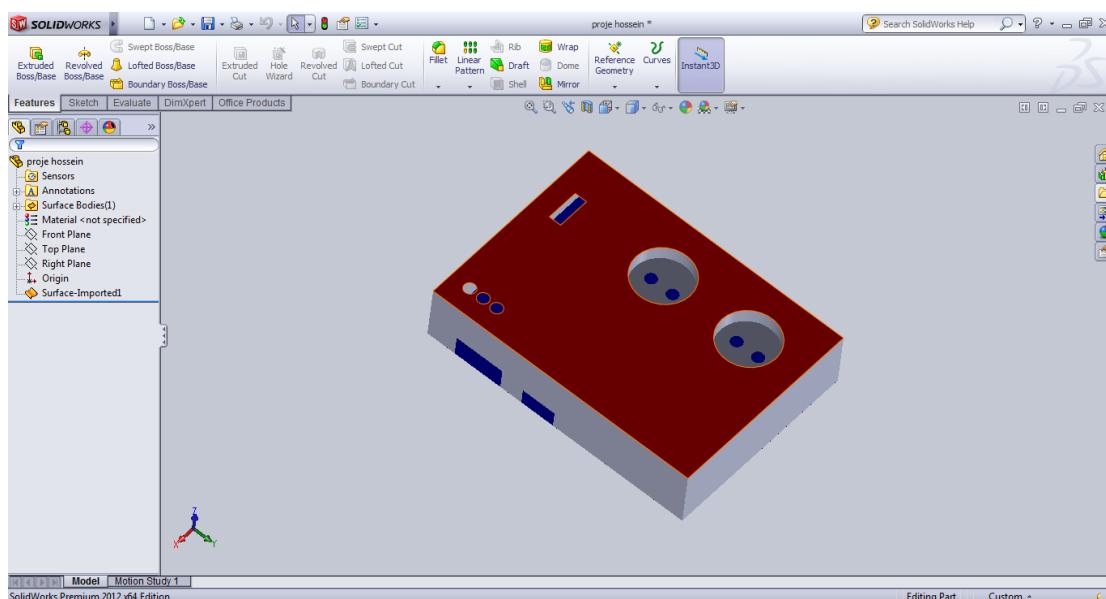
- Project View:** Shows a tree view of files: C:\Users\Administr... (main.py, project.md).
- Editor:** The main editor window displays the Python file `main.py` with the following code:

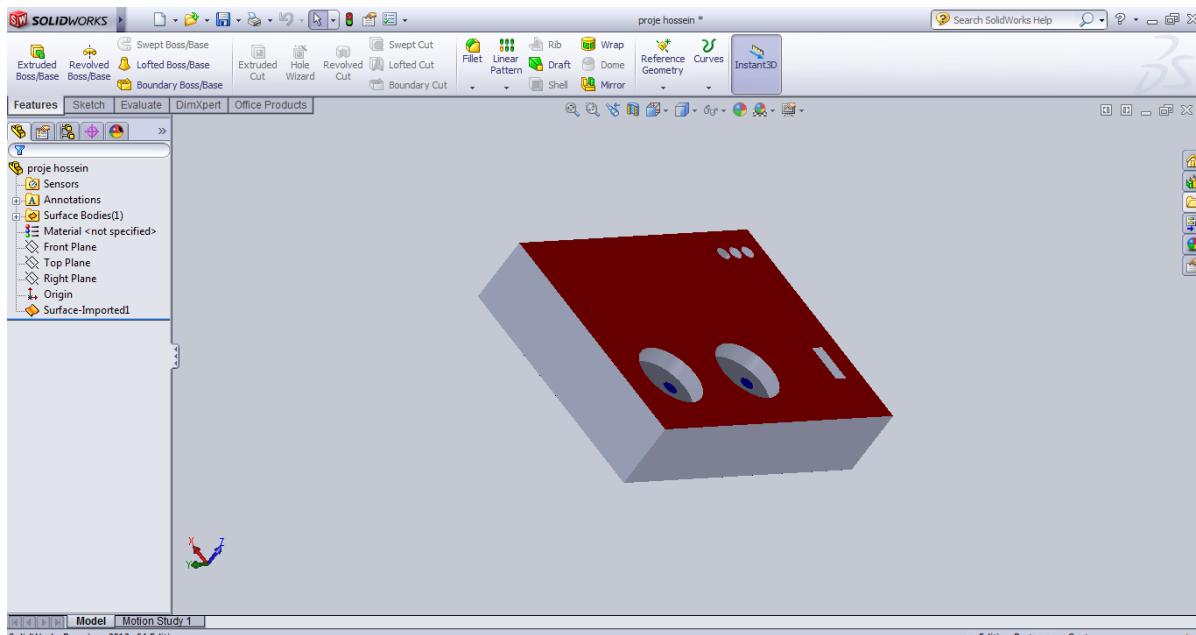
```
18 #!/usr/bin/python
19 # from broadcom.bcm43362 import bcm43362 as wifi_driver,
20 # from expressif.esp32net import esp32wifi as wifi_driver,
21 # from expressif.esp8266wifi import esp8266wifi as wifi_driver,
22 #
23 # init the wifi driver!
24 # The driver automatically registers itself to the wifi interface,
25 # with the correct configuration for the selected device.
26 wifi_driver.auto_init()
27 -
28 streams.serial()
29 -
30 -
31 # use the wifi interface to link to the Access Point.
32 # change network name, security and password as needed.
33 print("Establishing Link...")
34 try:
35     ... # FOR THIS EXAMPLE TO WORK, "Network-Name" AND "Wifi-Password" MUST BE SET.
36     ... # TO MATCH YOUR ACTUAL NETWORK CONFIGURATION.
37     wifiLink("Network-Name","wifI.WIFI_WPA2","Wifi-Password")
38 except Exception as e:
39     print("oops, something wrong while linking :(", e)
40     while True:
41         sleep(1000)
42     -
43     # define MQTT callbacks
44     def is_sample(data):
45         ... if ("message" in data):
46             return (data['message'].qos == 1 and data['message'].topic == "desktop/samples")
47         return False
48     -
49     def print_sample(client,data):
50         message = data['message']
51         message = data['message']
```
- Terminal:** The bottom terminal window shows the output of the command `textpython C:\Users\Administrator\MQTT_at_mosquitto_org\main.py`, which includes:

```
WARNING!1 Can't retrieve user info!
Writing C:\Users\Administrator\main.py
Writing C:\Users\Administrator\readme.md
[warning] Can't create remote project
Cloning From C:\Users\Administrator\zerynth2\dist\r2.0.11\libs\official\zerynth\mqtt\examples\MQTT_at_mosquitto.org
```

## پیوست ۴ طراحی جعبه

طراحی جعبه در دو مرحله انجام شد ، یک مرحله اقدام برای طراحی با Solidwork انجام گرفت، تا توسط پرینتر سه بعدی به چاپ برسد که تصاویر طرح را می‌توانید در زیر مشاهده کنید.





که به دلایل مختلفی که مهمترین آن اقتصادی نبودن آن ، می باشد. این طرح منتفی شد.

در گام بعدی به پیشنهاد استاد راهنمای ، از باکس آماده استفاده شد ، و قطعات و مدارات

درون آن مونتاژ شد.