



معرفی و بررسی پروتکل ارتباطی MQTT مخصوص اینترنت اشیا (IoT)

معرفی و بررسی پروتکل ارتباطی MQTT مخصوص اینترنت اشیا (IoT)

 MQTT



<https://blog.microele.com>

تاریخ انتشار ۴ دی، ۱۴۰۰ توسط محمد جواد رشیدیانفر

سلام خدمت همه شما میکروالکامی ها. در مطالب قبلی از [سری آموزش های ماژول SIM800](#) به بررسی کامل و نحوه [ارسال دیتا به سرور با استفاده از SIM800](#) پرداخته شد. همچنین در مطلبی دیگر نیز نحوه استفاده و [ارسال دیتا از طریق HTTP](#) مورد بررسی قرار گرفت. در این مطلب به بررسی و معرفی کامل پروتکل MQTT و معرفی چند Broker رایگان آن مخصوص اینترنت اشیا (IoT) پرداخته خواهد شد. پس با من تا انتهای مطلب همراه باشید. همچنین شما



میتوانید سایر مطالب من رو از [این لینک](#) مطالعه و بررسی کنید.

معرفی پروتکل MQTT

در بحث اینترنت اشیا (IoT) راه و روش های برقراری ارتباط یا همان پروتکل های پر مصرفی وجود دارد. یکی از معروف ترین این موارد MQTT می باشد. MQTT در سال 1999 توسط [Andy Stanford-Clark](#) و Arlen Nipper ساخته و معرفی شد. در ابتدا IBM از آن برای مانیتور و نظارت بر خطوط لوله نفت در سیستم کنترل صنعتی SCADA استفاده شد. می توان گفت پروتکل MQTT یک پروتکل برای کاربرد های (M2M (Machine to Machine) و IoT می باشد. نحوه ارتباط در این پروتکل بصورت کلاینت-سرور بوده و همچنین در لایه Network یا شبکه قرار دارد.

MQTT مخفف Message Queuing Telemetry Transport است. این پروتکل یک بستر و پروتکل ساده ای می باشد که روی TCP/IP سوار شده و تبادل دیتا را بر مبنای سوکت (Socket) انجام خواهد داد. نوع دیگر آن MQTT-SN است که برای تبادل دیتا از UDP یا بلوتوث استفاده می کند. می توان گفت دلیل اصلی سبک بودن و سریع بودن این پروتکل این است که هدر های موجود در این پروتکل دارای حجم کم بوده و لذا منجر به کاهش ترافیک خواهد شد.

علاوه بر این باعث سادگی در پیاده سازی این پروتکل نیز خواهد شد. همه این موارد مزیدی بر این علت است تا MQTT گزینه مناسبی جهت انتخاب و تبادل دیتا در اینترنت اشیا باشد. چرا که بدلیل محدود بودن سخت افزار میکروکنترلر ها براحتی می توان از آن استفاده کرد. معماری این پروتکل به شکل توپولوژی ستاره (Star) است که با یک یا چندین دستگاه و پابلیشر ارتباط برقرار می نماید.

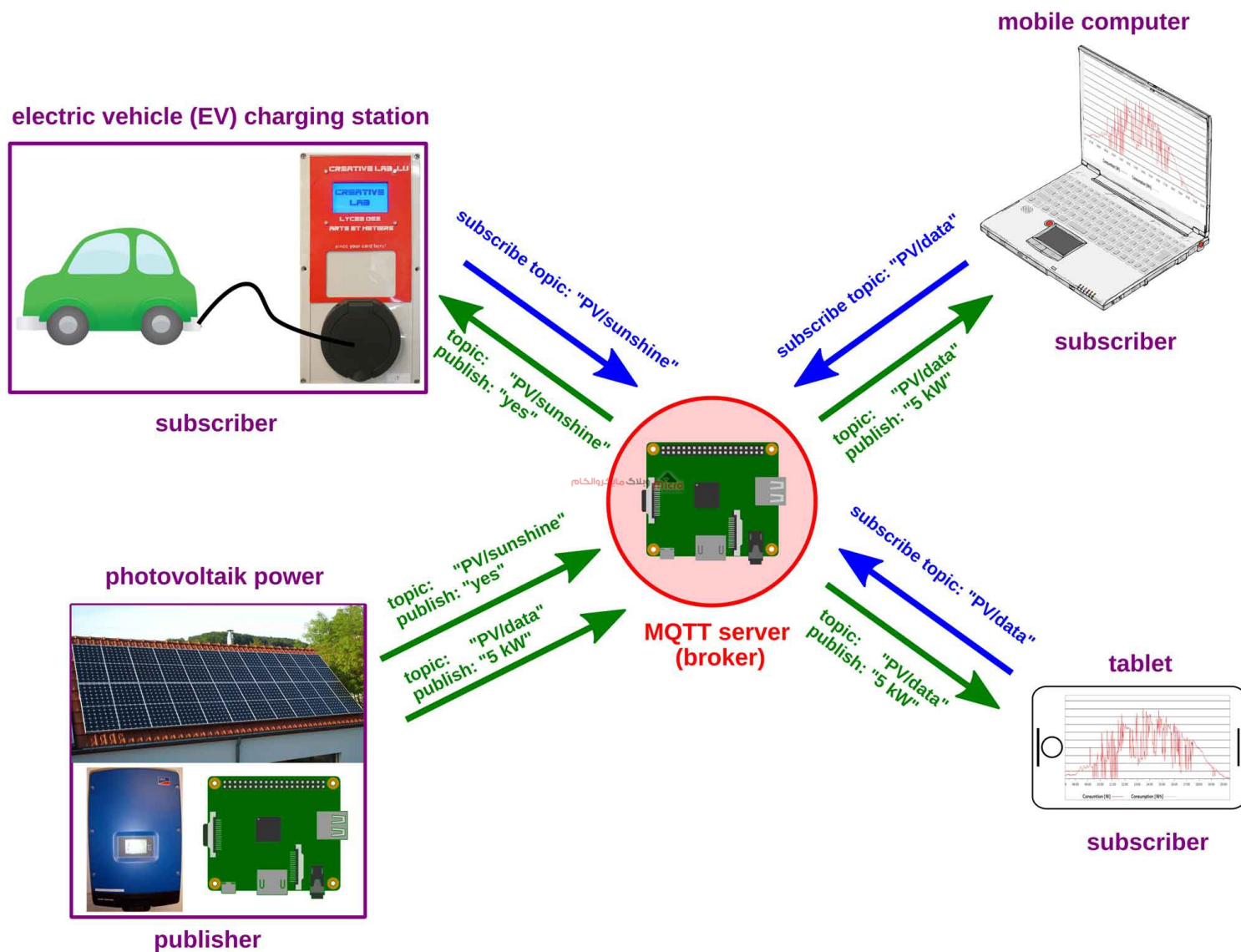
نحوه عملکرد و تبادل دیتا در MQTT

همانطور که مشخص است پروتکل های اینترنتی به یک سرور نیاز دارند تا دیتا ها را دریافت/ارسال کرده و پردازش های لازم را انجام دهند و نهایتا آن را ذخیر کنند. همین امر نیز در MQTT صدق می کند اما با فرایندی کمی متفاوت تر.

تصور کنید که شما یک سایت آموزشی راه اندازی نموده اید و یک سری دیتا و مطالب آموزشی برای کاربران روی آن قرار داده اید. در این حالت هر شخصی بخواهد می تواند از هر دیتایی استفاده نماید. در حقیقت این امر باعث می شود بار پردازشی کاهش یافته و کاربر و ارائه دهنده سریع تر به هدف اصلی خود نزدیک شوند. در MQTT هم دقیقا از روشی مشابه استفاده می شود.



در این مثال اینترنت نقش اینترنت و TPC/IP را ایفا کرده، سایت مورد نظر نقش Broker یا کارگزار در MQTT و دیتا ها و مطالبی که منتشر می شود کار دستگاه Publisher را ایفا خواهد کرد. در این پروتکل دیتا ها را در قالب یک متن ساده بدون استفاده از رمزنگاری تبادل می شوند. برای این مورد می توان از TLS برای رمزگذاری و محافظت از دیتا ها در مقابل رهگیری، اصلاح و تغییر یا جعل آنها اقدام کرد.



نحوه عملکرد MQTT



MQTT در Publisher

Publisher یا توزیع کننده در واقع همان دستگاه ما یا GateWay یا سنسور های ما است که یک Topic را در Broker ایجاد کرده و دیتا هایی که جمع آوری می شود را بر روی آن بارگزاری می کند. در بحث Topic هر دستگاه می تواند چند Topic داشته باشد و همزمان در Topic های دیگر نیز عضو شده و دیتا خود را نیز در آنجا منتشر کند. هر پابلیشر برای اینکه در یک Topic عضو شود باید آدرس منحصر به فرد آن را داشته باشد. در MQTT یک Topic می تواند ساختار درختی نیز داشته باشد. یعنی هر Topic نیز می تواند دارای چندین زیر Topic باشد.

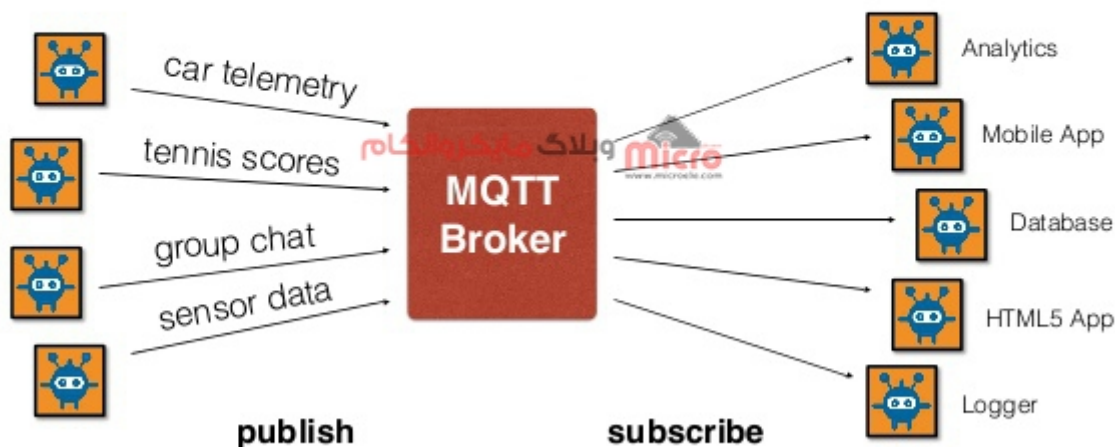
MQTT در Broker

بدیهی است برای پیاده سازی یک شبکه MQTT کامل و دریافت اطلاعات به Broker (براکر) نیاز است. در سراسر اینترنت Broker های متعددی وجود دارد که با استفاده از نصب آنها روی یک سرور شخصی این امکان را خواهید داشت که آن را به یک Broker شخصی و مخصوص کاربرد خود تبدیل نمایید. علاوه بر این مورد استفاده از برخی از Broker های رایگان نیز وجود دارد. طبیعتاً استفاده از دسته دوم خالی از برخی محدودیت ها نخواهد بود.



MQTT

pub/sub decouples **senders** from **receivers**



MQTT

آدرس دهی در MQTT

در MQTT برای آدرس دهی Topic ها باید اقدام کرد. ساختار آدرس دهی در این پروتکل مشابه لینوکس و بصورت درختی می باشد. با استفاده از مثال زیر در آدرس یاد شده عضو (Subscribe) می شویم و میزان رطوبت گلدان موجود در دفتر کار را خواهیم خواند.

```
sensors/humidity/office/flower-sensor
```

مطابق با مثال فوق با استفاده از "/" می توان ساختار درختی در آدرس دهی را پیاده سازی کرد. ساختار آدرس دهی در هر تاپیک توسط سنسور یا پابلیشر مشخص خواهد شد و یک چیز از پیش تعیین شده و استاندارد نیست که ساختار مشخصی داشته باشد. برای مشاهده تمام زیر تاپیک های مربوط به آدرس از دستور زیر استفاده می کنیم. با این کار Brocker زیر تاپیک های آدرس را در اختیار ما قرار خواهد داد.



sensors/humidity/office/#

MQTT در (Quality of Service (QoS

در MQTT مفهوم مهم دیگری تحت عنوان Quality of Service نیز وجود دارد که باعث می‌شود در واقع نحوه ارسال پیام‌ها را مدیریت نماییم. در واقع یک جور اولویت بندی می‌شود. در MQTT سه مدل QoS وجود دارد. فرض کنید چند سنسور دما-رطوبت داشته و سنسور n ام اگر دیتا هایش توسط قطع ارتباط به برآکر ارسال نشود زیاد مهم نباشد اما سنسور m ام این قضیه برایش خیلی مهم تر باشد. لذا در این حالت از QoS دیگری در ارسال دیتا استفاده می‌کنیم که امکان از بین رفتن دیتا در آن کمتر باشد.

QoS 0

QoS Level 0 یا همان QoS 0 ساده ترین حالت تبادل دیتا با broker می‌باشد. در این سطح به acknowledgment نیاز نمی‌باشد پس ارتباط ساده تر است. عدم نیاز به acknowledgment به معنای آن است که، اگر پیام توسط Broker دریافت شود یا نشود اهمیتی ندارد و تنها به اینکه acknowledgment بستر TCP دریافت کند فرض را بر این خواهد گذاشت که دیتا ارسال شده و آن را از لیست ارسال حذف خواهد کرد. بدیهی است در این روش اگر ارتباط قطع گردد ممکن است بخشی از دیتا ارسال به Broker نرسد.

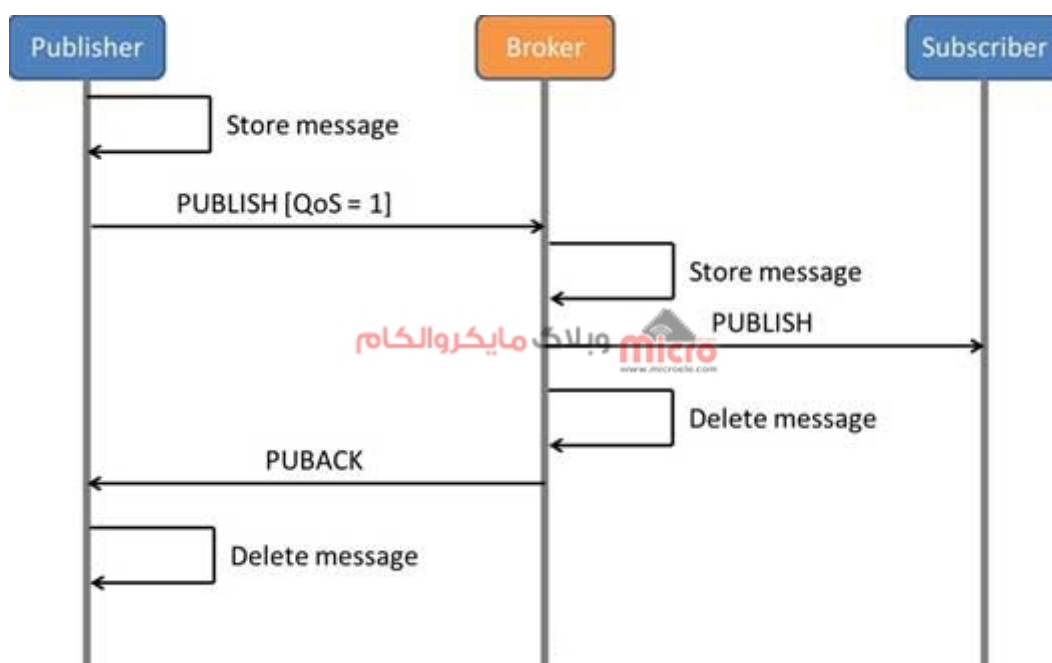


MQTT در QoS 0



QoS 1

در این سطح Publisher اطمینان حاصل خواهد شد که دیتا توسط Broker حداقل یک بار دریافت شده است. در تشریح عملکرد این سطح باید گفت Broker بعد از دریافت دیتا یا پیام یک پیامی تحت عنوان PUBACK به پابلیشر ارسال خواهد کرد. پابلیشر با دریافت این پیام اطمینان حاصل خواهد کرد که پیام مورد نظر توسط Broker دریافت شده است و لذا پیام از صف ارسال پیام ها خارج و حذف خواهد شد. چنانچه PUBACK توسط پابلیشر دریافت نشود مجدداً پیام به Broker ارسال خواهد شد. این روند تا زمانی که PUBACK دریافت شود ادامه پیدا خواهد کرد تا بالاخره دیتا به Broker برسد.



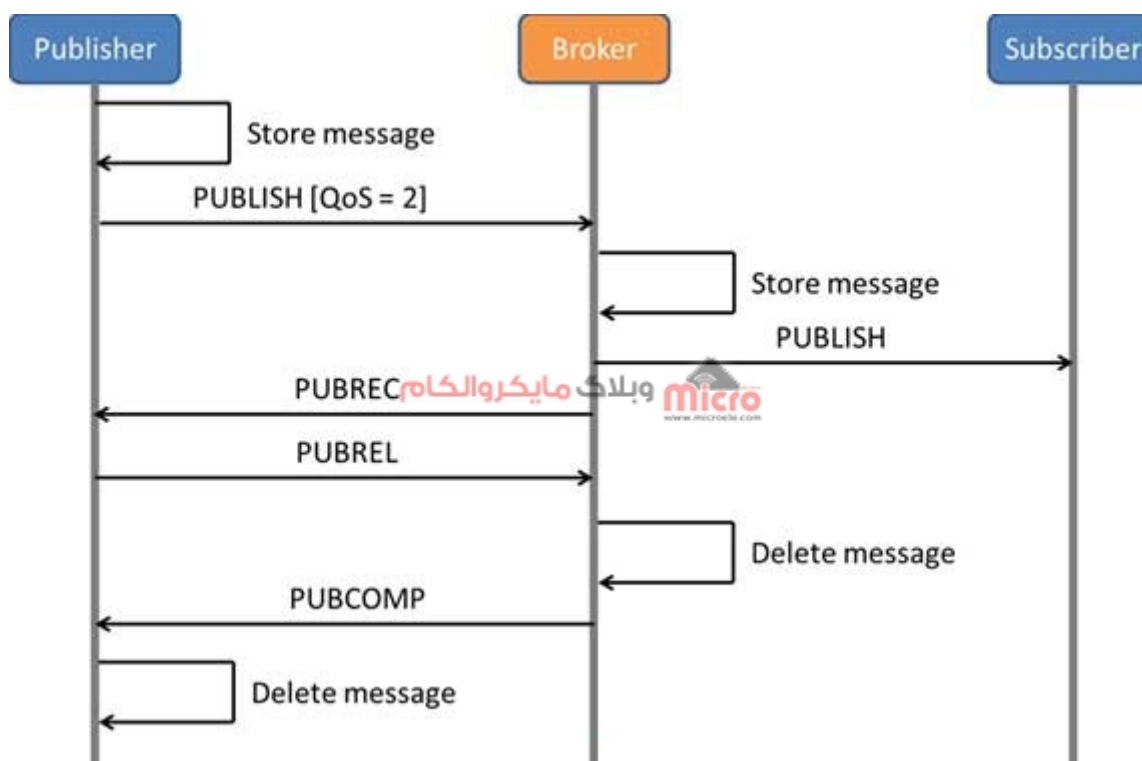
MQTT در QoS 1

QoS 2

در سطح QoS 2 ارسال کننده پیام یا همان پابلیشر از اینکه پیام تنها فقط یک بار نه بیشتر توسط یک Subscriber دریافت شده است اطمینان حاصل می‌کند. با استفاده از این حالت در موارد فوق حساس و مهم می‌توانیم مطمئن شویم که دستگاه دیتا های ارسال را دریافت کرده است. باتوجه به توضیحات ذکر شده بدیهی است که ترافیک بیشتری بین Publisher و Broker رد و بدل خواهد شد. این امر منجر به این خواهد شد که دیتا ارسال فقط یک بار دریافت شده است



و نه بیشتر از آن.



MQTT در QoS 2

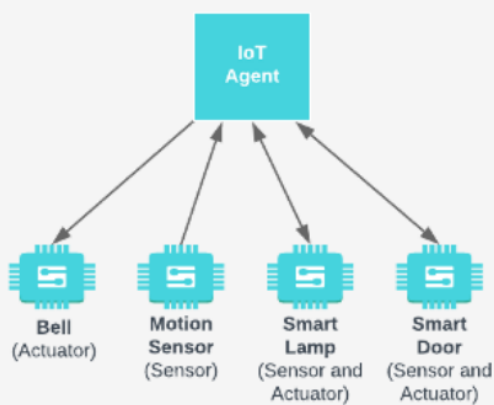
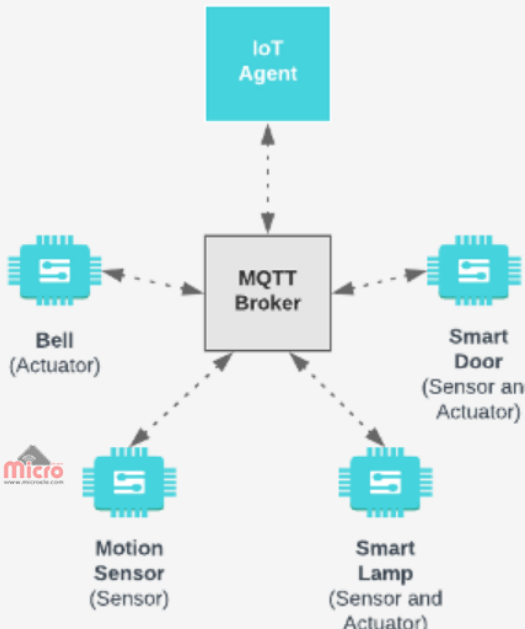
مقایسه با HTTP

اگرچه HTTP در وب دارای محبوبیت زیادی است ولی انتخاب مناسبی برای پکت های کوچک و کم حجم در اینترنت اشیا (IoT) نمی باشد. HTTP سند محور (document-centric) بوده درحالی که MQTT دیتا محور (data-centric) می باشد. HTTP بر پایه متن است و تجزیه و تحلیل آن پیچیده تر خواهد بود. MQTT بصورت باینری بوده و عملیات رمزگذاری و رمزگشایی آن به مراتب ساده تر و آسان تر می باشد.

بدلیل مدل انتشار-اشتراک یا publish-subscribe در MQTT توزیع دیتا در این پروتکل بصورت یک به چند (one-to-many) است درحالی که در HTTP بصورت یک به یک محدود شده است. در MQTT هدر پیام Publish بصورت 24 بایت، پیام



Connect بصورت 14 بایت اما هدرها در HTTP می‌تواند تا 1 کیلو بایت نیز باشد.

HTTP Transport	MQTT Transport
	
IoT Agent communicates with IoT devices directly	IoT Agent communicates with IoT devices indirectly via an MQTT Broker
Request-Response Paradigm	Publish-Subscribe Paradigm
IoT Devices must always be ready to receive communication	IoT Devices choose when to receive communication
Higher Power Requirement	Low Power Requirement

مقایسه MQTT با HTTP

جریان مصرفی

پروتکل MQTT با وجود قابلیت ارسال و تبادل دیتای بیشتر در ساعت و همچنین با قابلیت اعتماد بیشتر نسبت به HTTP،



توان و جریان الکتریکی کمتری مصرف می‌کند. شایان ذکر است MQTT برای آغاز و آماده سازی ارتباط میزان مصرف جریان بیشتری دارد اما با باز ماندن این ارتباط به سرعت این مورد جبران خواهد شد. باید به این مورد اشاره کرد که جایگزین های بهتری بجای HTTP وجود دارد که با MQTT مقایسه گردد. برخی از این موارد مانند AMQP, CoAP, DDS, XMPP و STOMP می‌باشد.

	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.43%	16.13%	3.45%	4.23%
Messages / Hour	1708	160278	3628	263314
% Battery / Message	0.01709	0.00010	0.00095	0.00002
Messages Received	240 / 1024	1024 / 1024	524 / 1024	1024 / 1024

جریان مصرفی MQTT و سایر پروتکل ها

محدودیت هایی که باید در استفاده از MQTT در نظر گرفت

- ساختار دیتا های منتشر شده برای مشترک (subscriber) مشخص نیست. در واقع این اطلاعات باید از قبل به اشتراک گذاشته شود یا به نوعی شناخته شده باشد. هنگامی که از فیلترینگ بر اساس topic ها استفاده شود



- publisher و subscriber باید topic صحیح را بدانند.
- تحویل دیتا و پیام تایید نمی‌شود. چرا که هیچ تاییدیه خودکار از طرف گیرنده وجود ندارد. علاوه بر این publisher همچنین نمی‌داند آیا مشترک (subscribe) وجود دارد یا نه.
- MQTT از پروتکل TCP استفاده می‌نماید که نیاز به handshaking پیچیده تری دارد و از نظر حافظه و توان مورد نیاز آسان نیست. طبق استاندارد می‌توان اینگونه برداشت کرد که TCP یک بستر مناسب انتقال است اما UDP اینگونه نیست. همچنین MQTT می‌تواند از TLS یا Websocket استفاده نماید.
- Broker بصورت متمرکز است و به نوعی قابلیت انعطاف پذیری و مقایس پذیری را از آن سلب نموده است.

معرفی Broker های MQTT

Broker های رایگان مخصوص MQTT کم نیستند که با [سرچ ساده در گوگل](#) به لیست بلند بالایی از آنها دست خواهید یافت. در ادامه چند مورد از آنها معرفی خواهد شد.

[Cloud MQTT](#)

این مورد از سرویس ابری آمازون استفاده کرده و MQTT را هم به شکل ساده و هم به شکل رمز شده ساپورت می‌نماید. شایان ذکر است این سرویس تا اتصال 10 دستگاه را رایگان پشتیبانی می‌کند اما برای دستگاه های بیشتر باید از اکانت غیر رایگان استفاده نمایید.

[HiveMQ](#)

این سرویس خدماتی رایگان را در MQTT عرضه می‌کند که دارای کیفیت و سرعت بالایی نیز می‌باشد. سایر قابلیت ها و سرویس های دیگر این سایت قابل خرید است.

برای آموزش استفاده از بروکر و کنترل GPIO های ماژول ESP8266/ESP32 و نحوه تبادل دیتا با استفاده از MQTT، می‌توانید [این مطلب](#) سایت را مطالعه نمایید.

[Eclipse IOT](#)

این سرورس پروتکل های MQTT و CoAP را پشتیبانی کرده و می‌توان رایگان از آن استفاده کرد. همچنین TLS نیز قابل



استفاده است و باعث می شود که دیتا بین Broker و دستگاه fw,vj رمزنگاری نگاری شده تبادل شوند.

نتیجه گیری

در این مطلب به معرفی کامل و نحوه عملکرد دستگاه در MQTT پرداخته شد. همانطور که بیان شد بدلیل سبک بودن این پروتکل گزینه ای مناسبی برای استفاده در اینترنت اشیا علی الخصوص در سیستم های مبتنی بر میکروکنترلر ها که دارای ظرفیت های رم و حافظه محدود تری هستند می باشد.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورت داشتن هرگونه نظر یا سوال درباره این آموزش اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتما به اشتراک بگذارید. همینطور میتونید این آموزش را پس از اجرای عملی توی اینستاگرام با هشتگ #microelecom به اشتراک بگذارید و **پیج مایکروالکام** (@microelecom) رو هم منشن کنید.