

Articles from FreeSandal

○ 《止》何處！？

2015-04-23 06:04:02 懸鉤子



止

《說文解字》：止，下基也。象艸木出有址，故以止爲足。凡止之屬皆从止。

《大學》講：知止而後有定，定而後能靜，靜而後能安，安而後能慮，慮而後能得。

一件事如果沒有它發生之理，那它能夠發生的嗎？假使那事果然都不發生，又如何能夠得到那個發生之理？這就是「事、理不二」的道理，發生之事蘊有發生之理，發生之理緣起發生之事。那「空山樹倒」是否是發生了一件事呢？假使有人砍了你的「櫻桃樹」又是不是發生了一件事的呢？

《易繫辭》說：艮，東北之卦也，萬物之所成終而所成始也，故曰成言乎艮。又說：終萬物始萬物者莫盛乎艮。

艮：艮其背，不獲其身，行其庭，不見其人，無咎。

象曰：艮，止也。時止則止，時行則行，動靜不失其時，其道光明。艮其止，止其所也。上下敵應，不相與也。是以不獲其身，行其庭不見其人，無咎也。

象曰：兼山，艮；君子以思不出其位。

初六：艮其趾，無咎，利永貞。

象曰：艮其趾，未失正也。

六二：艮其腓，不拯其隨，其心不快。

象曰：不拯其隨，未退聽也。

九三：艮其限，列其夤，厲薰心。

象曰：艮其限，危薰心也。

六四：艮其身，無咎。

象曰：艮其身，止諸躬也。

六五：艮其輔，言有序，悔亡。

象曰：艮其輔，以中正也。

上九：敦艮，吉。

象曰：敦艮之吉，以厚終也。

論語《里仁》子曰：「參乎！吾道一以貫之。」曾子曰：「唯。」子出。門人

問曰：『何謂也？』曾子曰：『夫子之道，忠恕而已矣。』

以及

老子第三十九章中講：

昔之得一者，天得一以清，地得一以寧，神得一以靈，谷得一以盈，萬物得一以生，侯王得一以為天下貞。其致之。天無以清則恐裂，地無以寧則恐發，神無以靈則恐歇，谷無以盈則恐竭，萬物無以生則恐滅，侯王無以貞高則恐蹶。故貴以賤為本，高以下為基。是以侯王自謂孤寡不穀，此非以賤為本耶？非乎？人之所惡，唯孤寡不穀，而侯王以為稱。故致譽無譽。不欲碌碌如玉，珞珞如石。

正是《止》於《一》之『學問』大道。希望人們知道所謂『道德』之名，實在說的是『得到』——得道——的啊！！

如果說『氣候』持續變遷，有朝一日，假使乾坤都『無路』可走，人又該往向『何方』的呢？？

既然已經談了『WebIOPi』，就讓我們也略說說『MQTT』吧：

《 MQTT.org 》

Frequently Asked Questions

What is MQTT?

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging “**machine-to-machine**” (M2M) or “**Internet of Things**” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium.

Who invented MQTT?

MQTT was invented by Dr Andy Stanford-Clark of IBM, and Arlen Nipper of Arcom (now Eurotech), in 1999.

Where is MQTT in use?

MQTT has been widely implemented across a variety of industries since 1999. A few of the more interesting examples are listed on the Projects page.

Is MQTT a standard?

As of March 2013, MQTT is in the process of undergoing standardisation at OASIS.

The protocol specification has been openly published with a royalty-free license for many years, and companies such as Eurotech (formerly known as Arcom) have implemented the protocol in their products.

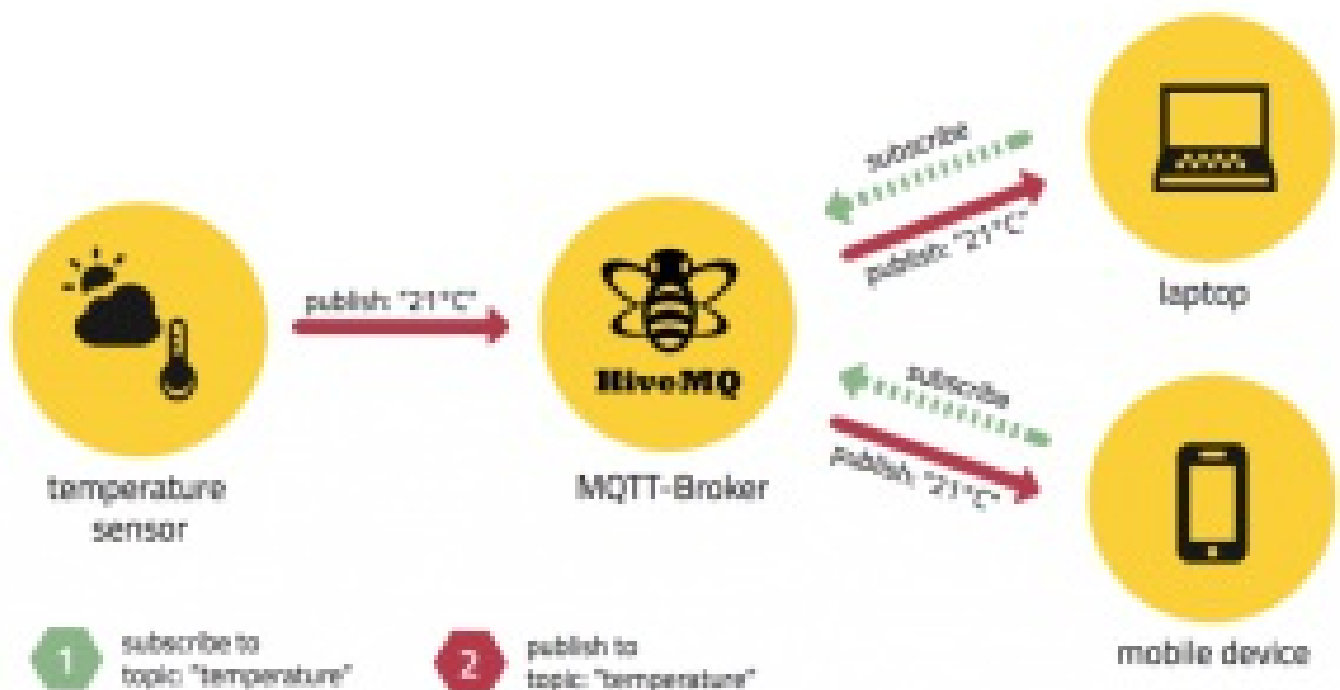
In November 2011 IBM and Eurotech announced their joint participation in the Eclipse M2M Industry Working Group and donation of MQTT code to

the proposed Eclipse Paho project.

《 **MQTT 101 – How to Get Started with the lightweight IoT Protocol** 》一文，詳談了「MQTT」的始末：

MQTT was developed by Andy Stanford-Clark (IBM) and Arlen Nipper (Eurotech; now Cirrus Link) in 1999 for **the monitoring of an oil pipeline through the desert. The goals were to have a protocol, which is bandwidth-efficient and uses little battery power, because the devices were connected via satellite link and this was extremely expensive at that time.**

The protocol uses a publish/subscribe architecture in contrast to HTTP with its request/response paradigm. Publish/Subscribe is **event-driven** and enables messages to be pushed to clients. The central communication point is **the MQTT broker**, it is in charge of dispatching all messages between the senders and the rightful receivers. Each client that publishes a message to the broker, includes a topic into the message. The topic is the routing information for the broker. Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client. **Therefore the clients don't have to know each other, they only communicate over the topic.** This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers.

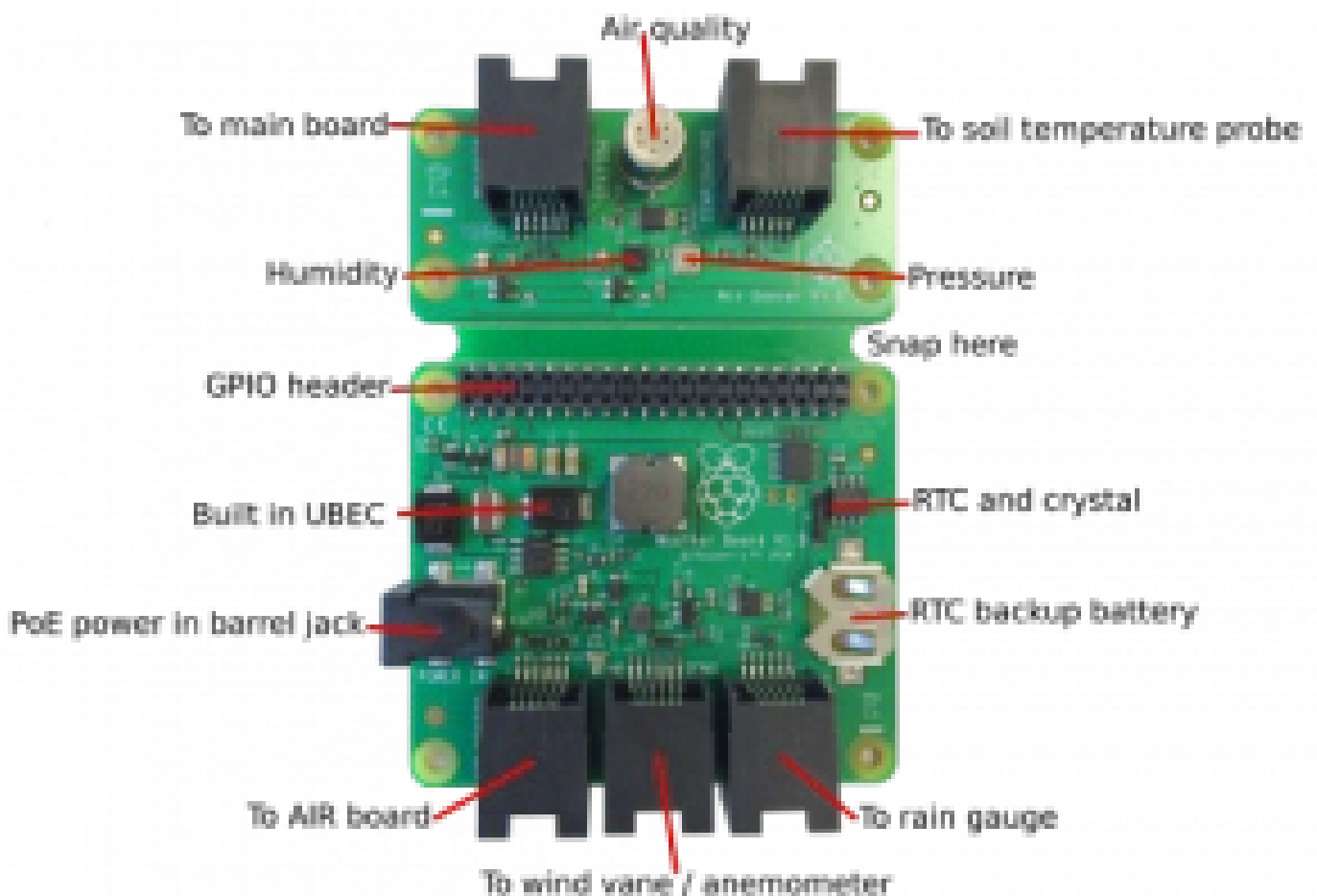


The difference to HTTP is that a client doesn't have to pull the information it needs, but the broker pushes the information to the client, in the case there is something new. Therefore each MQTT client has a permanently open TCP connection to the broker. If this connection is interrupted by any circumstances, the MQTT broker can buffer all messages

and send them to the client when it is back online.

As mentioned before the central concept in MQTT to dispatch messages are topics. **A topic is a simple string that can have more hierarchy levels, which are separated by a slash.** A sample topic for sending temperature data of the living room could be house/living-room/temperature. **On one hand the client can subscribe to the exact topic or on the other hand use a wildcard.** The subscription to house/+/temperature would result in all message send to the previously mention topic house/living-room/temperature as well as any topic with an arbitrary value in the place of living room, for example house/kitchen/temperature. The plus sign is a single level wild card and only allows arbitrary values for one hierarchy. **If you need to subscribe to more than one level, for example to the entire subtree, there is also a multilevel wildcard (#).** It allows to subscribe to all underlying hierarchy levels. For example house/# is subscribing to all topics beginning with house.

假使我們仔細閱讀《 [Raspberry Pi Weather Station for schools](#) 》文章的內容，然後「思考」與「想像」，成千上萬個「微氣象站」，散佈於「景點」、「學校」、「社區」與「家庭」……



，有著各種「感測器」

- Rainfall
- Wind speed
- Wind gust speed

- Wind direction
- Ambient temperature
- Soil temperature
- Barometric pressure
- Relative humidity
- Air Quality
- Real Time Clock (for data logging purposes)

『產生』、『累積』和『分享』……的『大數據』，對於

- 生活便利
- 環境認知
- 科學研究
- 技術應用

.....

將有深遠的影響，也許能為『未來』帶來《曙光》，建立人類自身的《環中基地》，那時就《樹莓派真實可吃》的了！！

意猶未盡者，可以接著瀏覽

- 《 [IBM Internet of Things Foundation \(IoT Foundation\)](#) 》
- 《 [Use a Raspberry Pi to connect to the IBM Internet of Things Foundation.](#) 》
- 《 [MQTT Servers/Brokers](#) 》
- 《 [paho-mqtt 1.1](#) 》