事例紹介
「IoTと情報セキュリティの課題」

ソフトウェアテストシンポジウム 2018 新潟
JaSST'18 Niigata
開催　2018年4月20日(金)

今岡工学事務所
今岡　通博

https://www.facebook.com/imaoka.micihihiro
imaoca@gmail.com,@imaoca
http://www.itmedia.co.jp/author/208685/
https://fpgastartup.connpass.com/
https://blogs.msdn.microsoft.com/devwire/

今岡　通博

imaoka michihiro

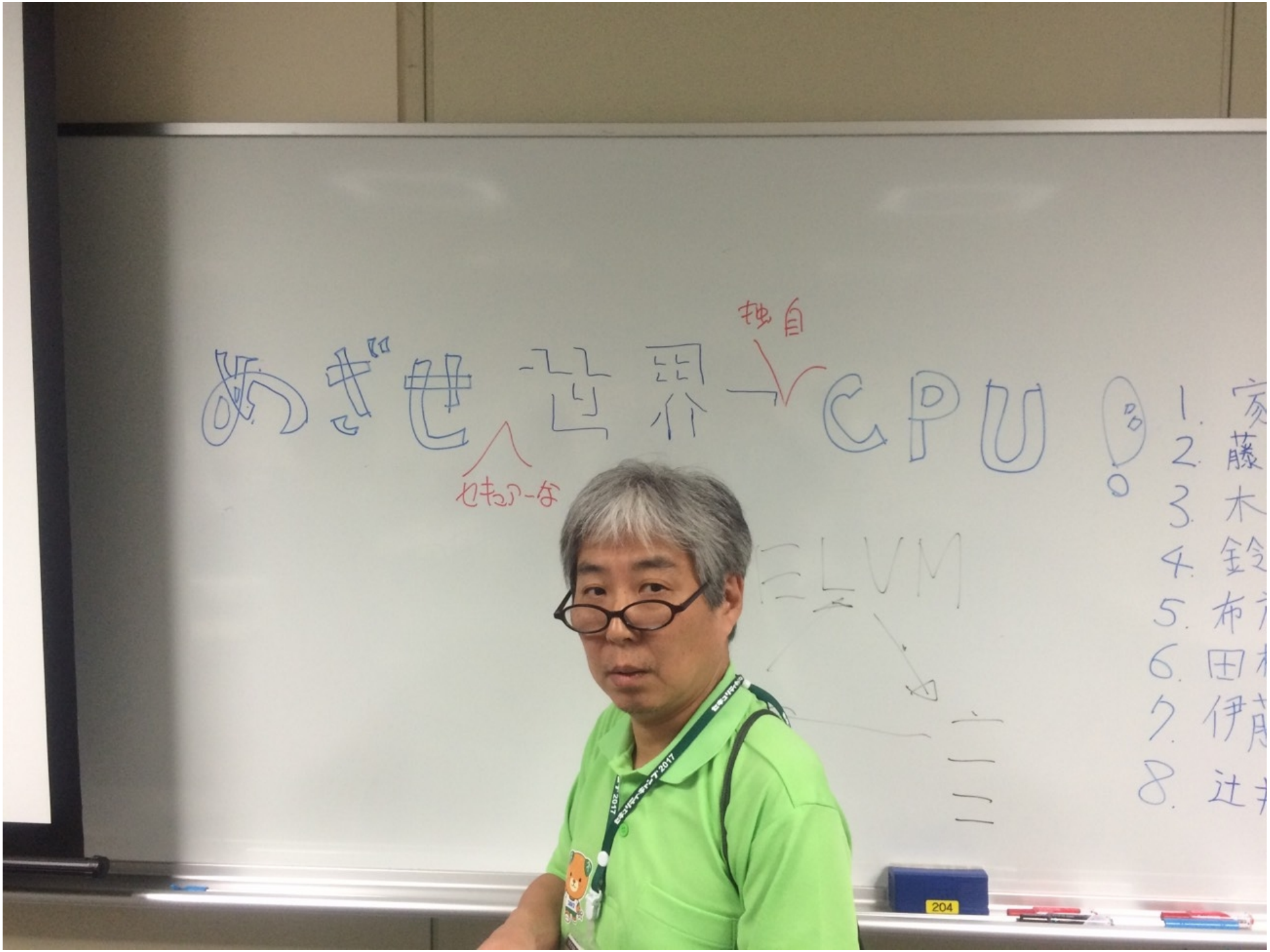
@imaoca

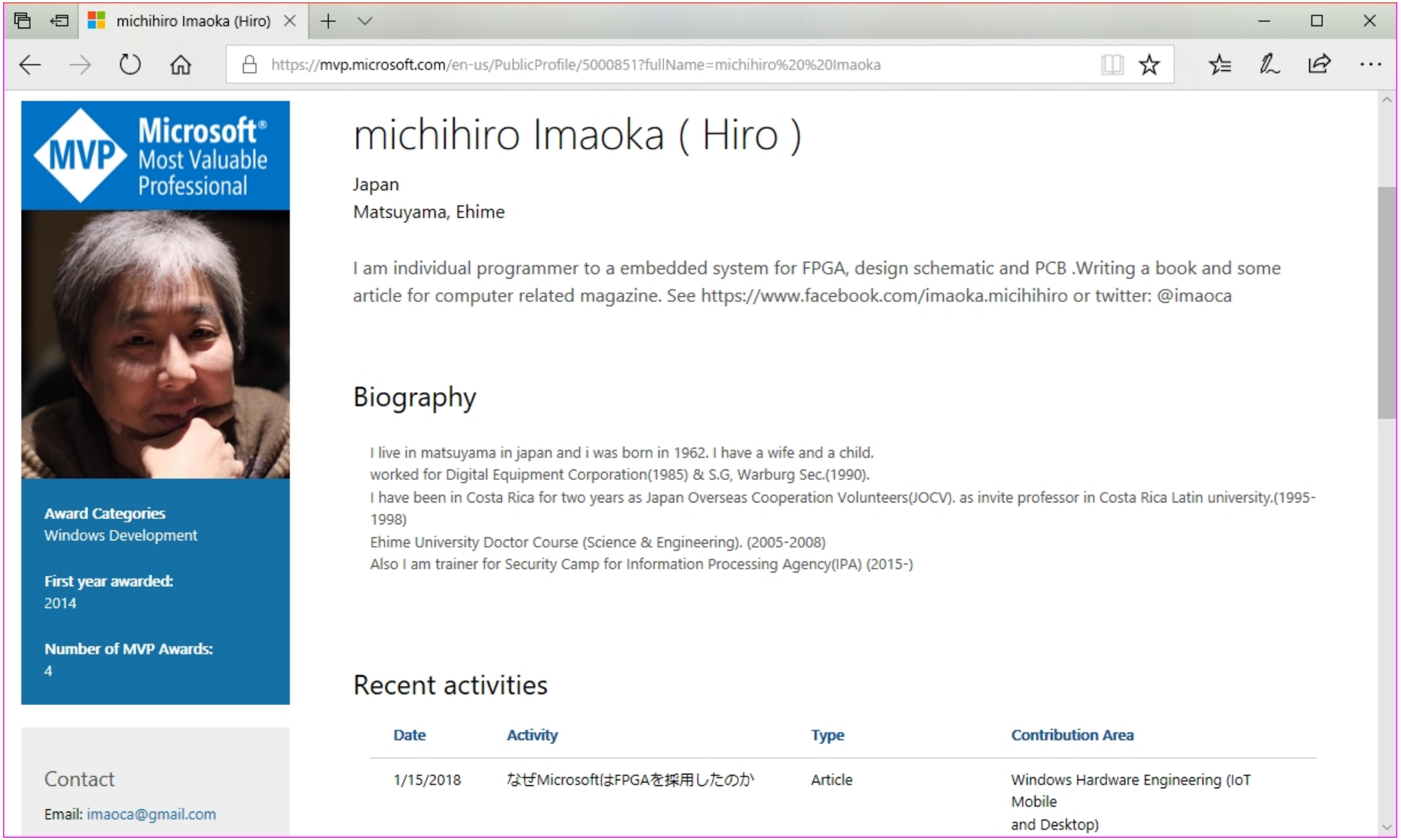
今岡工学事務所　個人事業主代表

組込み系受託開発、FPGA応用製品、書籍等執筆、回路設計、基板設計、非常勤講師

愛媛県松山市在住

・元日本DEC(Digital Equipment Corp.)他
・最近は原稿を書いていることが多い
・セキュリティ・キャンプの講師とか
・ハードウエアスタートアップ支援
・Microsoft MVP(Windows Development)
・FPGAスタートアップ主宰
・Androidの会コミュニティ運営委員
・愛媛県警察部外通訳員（英語、西語）


独自
CPU
LLVM
1.
2.
3.
4.
5.
6.
7.
8.
204



https://mvp.microsoft.com/ja-jp/PublicProfile/5000851
imaoca@gmail.com
twitter: @imaoca
https://www.facebook.com/imaoka.micihihiro

今岡工学事務所船籍のヨット２艇

夢は海上移動オフィス

セッションの内容

・バズワードとなって久しいIoTですが、日本語では「もののインターネット」と訳されて認知されてはいるものの、その実もっともとらえどころのない技術用語のひとつとなっています。

・IoTを支える基盤は組込技術、センシング技術、情報通信技術、クラウド技術の上に成り立っています。これらの複合技術の上に成り立っているテクノロジーゆえすべてを俯瞰的に見通せるスキームが確立されていません。

・今回はIoTの本質と、組込開発におけるセキュリティを担保するためのテスト等について事例を交えこれらの課題について議論したいと思います。

IoTの本質

IoTにおけるセキュリティを担保するためのテスト

IoTにおけるセキュリティ

・データの改ざん
・なりすまし
・乗っ取り、踏み台
・マルウエア
・通信の傍受
・通信妨害

今回は通信妨害について
考えます

通信妨害に関して
IoTのプロトコルとしてディファクト
になりそうなMQTTというプロトコルを
事例に考えます。

IoTにおけるMQTT のユースケース

インターネットにつながったエンドノードに接続されたLEDをスマートフォンから制御します。

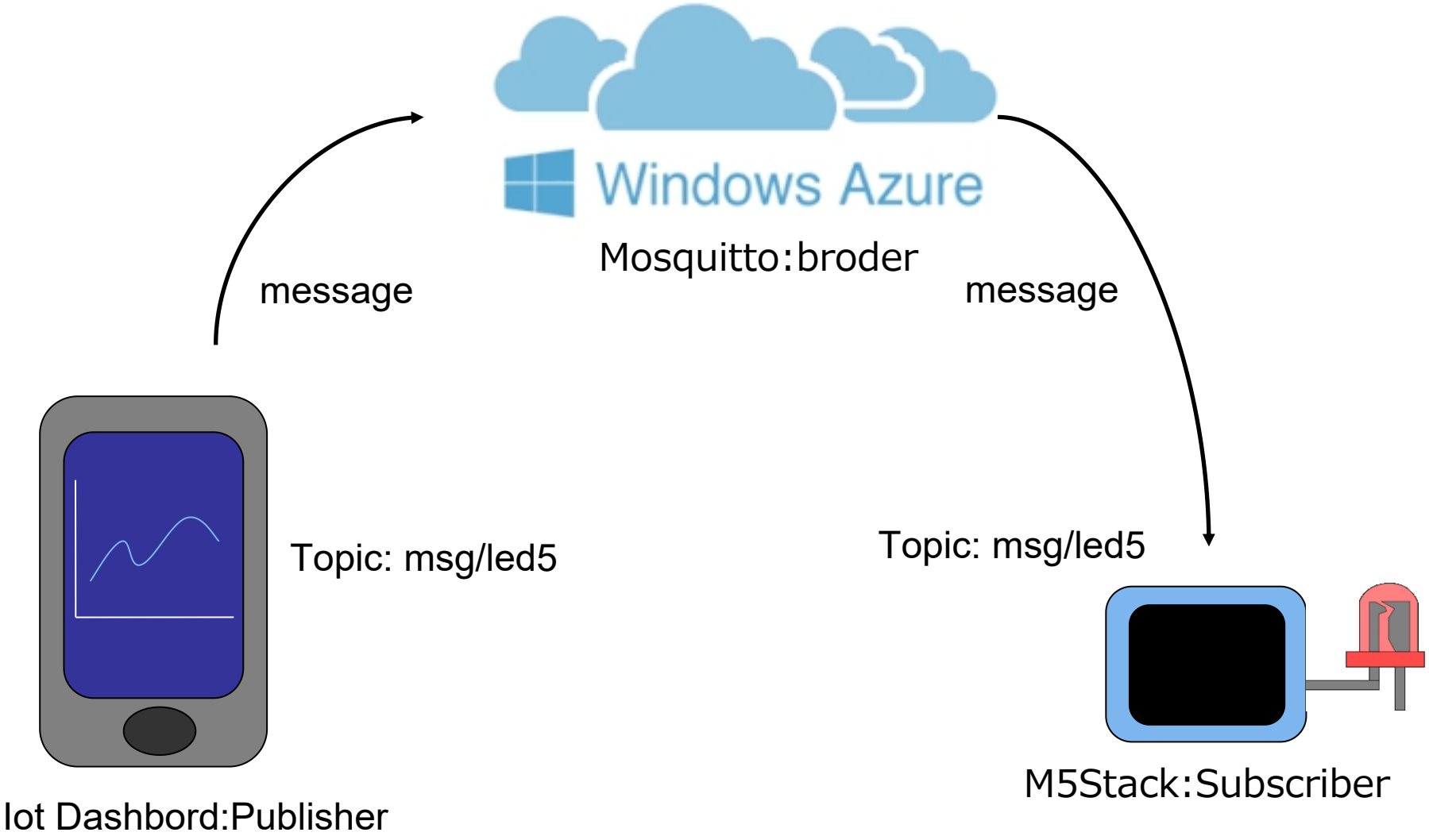
この仕組みを応用すればインターネットにつながった世界中の機器を電気的に制御可能であることを示唆します。

もっともシンプルなIoT実装モデル。

いくつかの単語

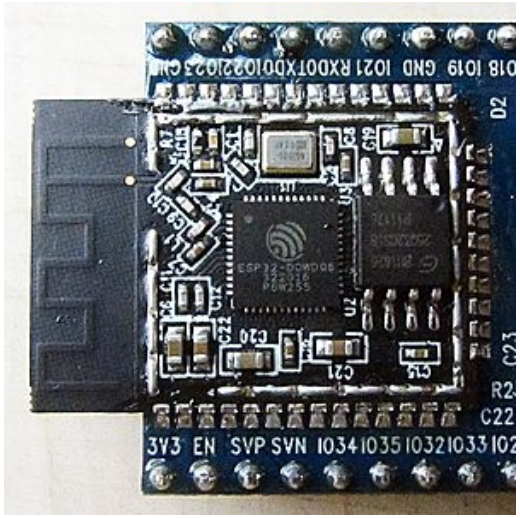
エンドノードにはM5STACKを使用
接続プロトコルはMQTT
スマホではIoT Dashboardを使用
BrokerはMosquitto
MessageはBrokerを介してSubscriberに送られる
Subscriberは受け取りたいMessageのTopicを指定

デモの構成


Windows Azure
Mosquitto:broder
message
message
Topic: msg/led5
Topic: msg/led5
Iot Dashbord:Publisher
M5Stack:Subscriber

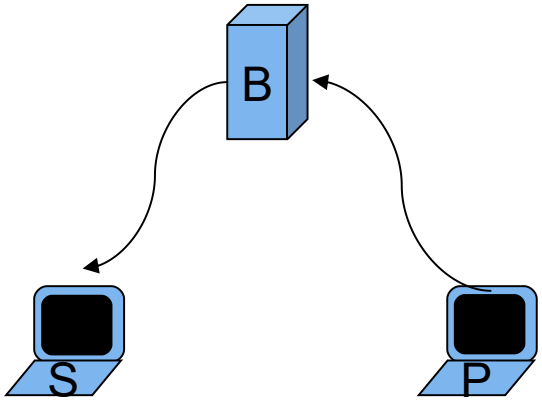
M5Stackとは

ESP32が入っています。
　　WiFi とBluetoothの機能
　　Tensilica の32bit ２コアCPU
　　メモリー:520KiB SRAM
　　AD,DA,SPI,I2C,UART,GPIO
カラーLCD内蔵（320X240）
Arduinoの開発環境を流用
中国から買えば＄31だった
Microsoft Azure Certified
詳しくはここを見てください
http://m5stack.com/



MQTTとは

Message Queuing Telemetry Transportの略
MQTTはM2MやIoTのための接続プロトコルで
　Publisher,Broker,そしてSubscriberで構成
軽量で高速（と言われている）
TCPより上のレイヤー
http://mqtt.org/

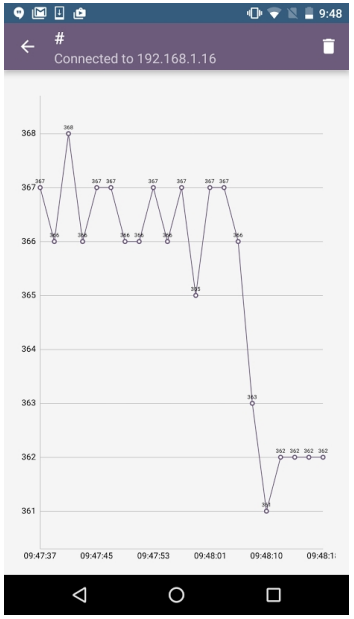


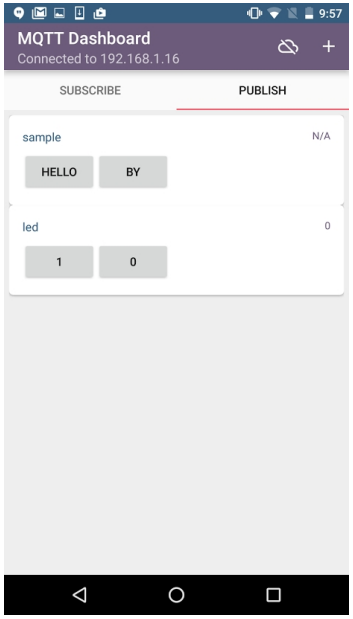
Mosquittoとは

MosquittoはMQTTのひとつの実装
もっともMQTT仕様書に忠実な実装で他の実装のリファレンスとなる
Broker,Client(Publisher, Subscriber)ともオープンソース
ポートは通常1883
https://mosquitto.org/

IoT Mqtt Dashboard

Androidで動作するMQTTクライアントアプリ
publisherとsubscriberとして動作可
簡単な操作でボタンなどのGUIを作成可
メッセージが数値の場合グラフ表示が可

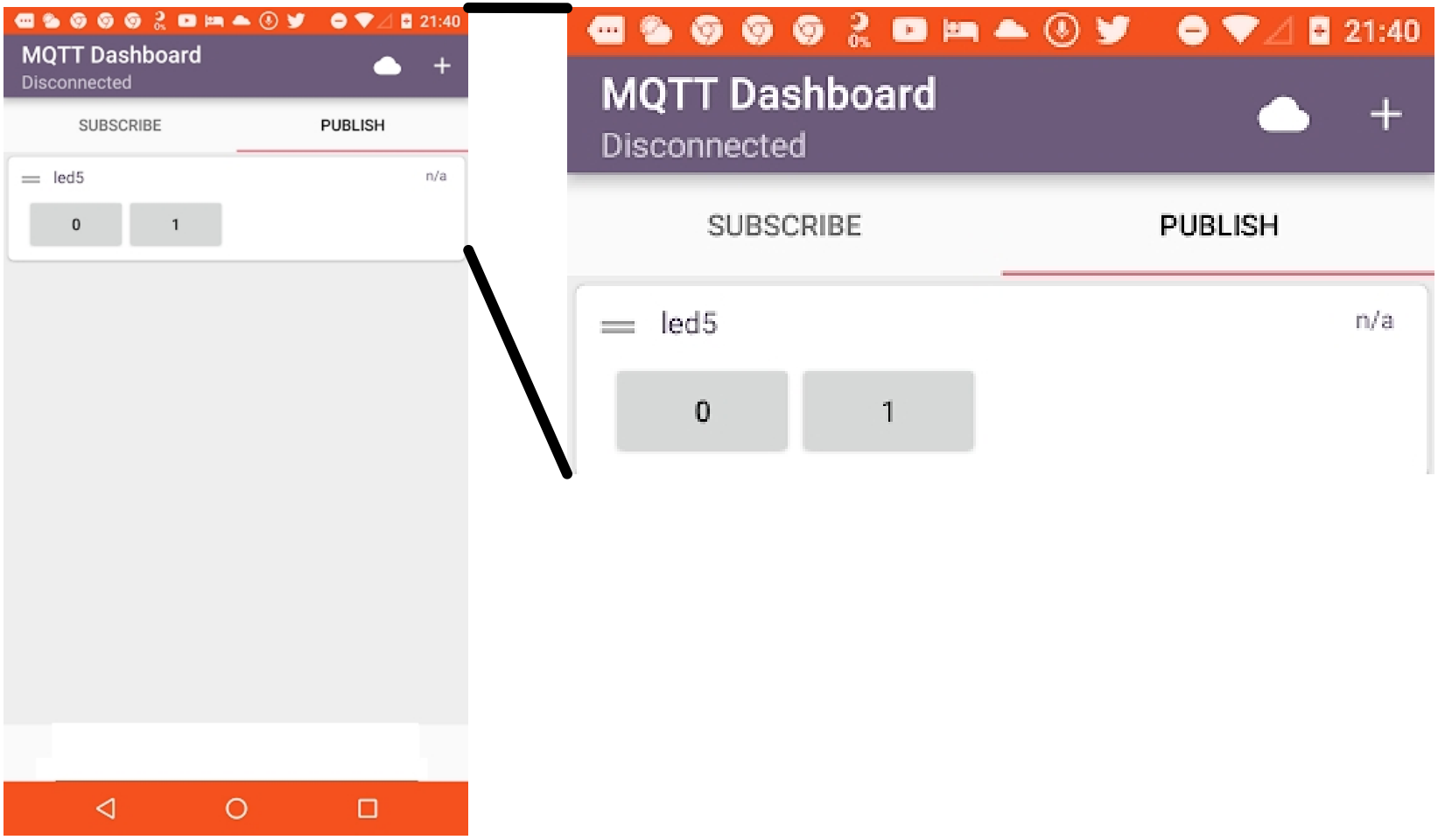




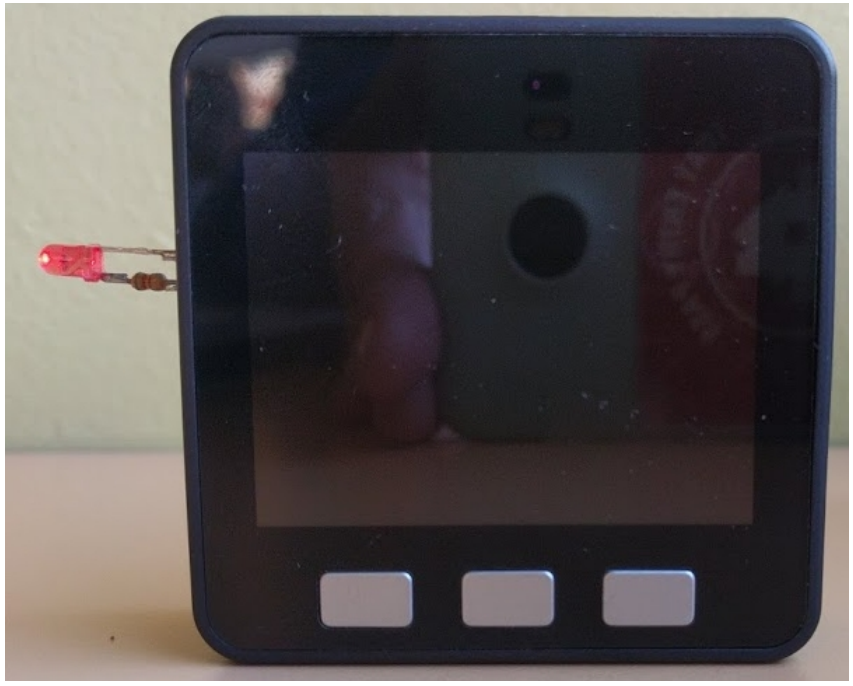
M5Stackのプログラミング

最新版は以下のgithubで公開予定　https://github.com/imaoca

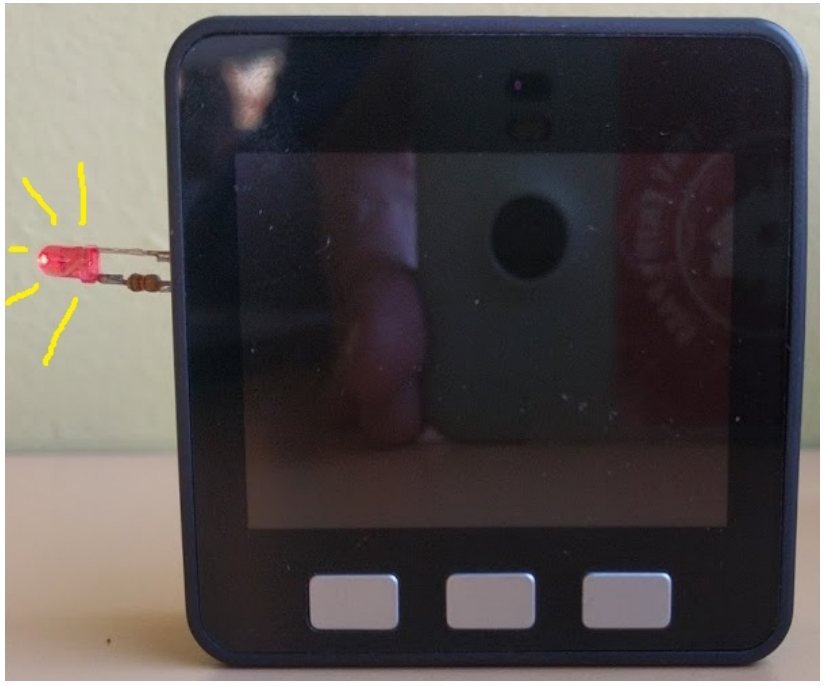
スマホからLEDを制御する画面



M5StackのLEDが点灯の様子



メッセージ"0"を受信



メッセージ"1"を受信

通信妨害のテスト

MQTTはIoTのプロトコルに適しており、不安定な回線や通信妨害に強いプロトコルだとされています。特に通信の品質を保証するQoSを何段階かのレベルで設定できるようになっており、通信妨害の耐性およびデータの重要性に応じて選択が可能です。

今回のテストではQoSレベルと通信妨害の耐性について観察します。

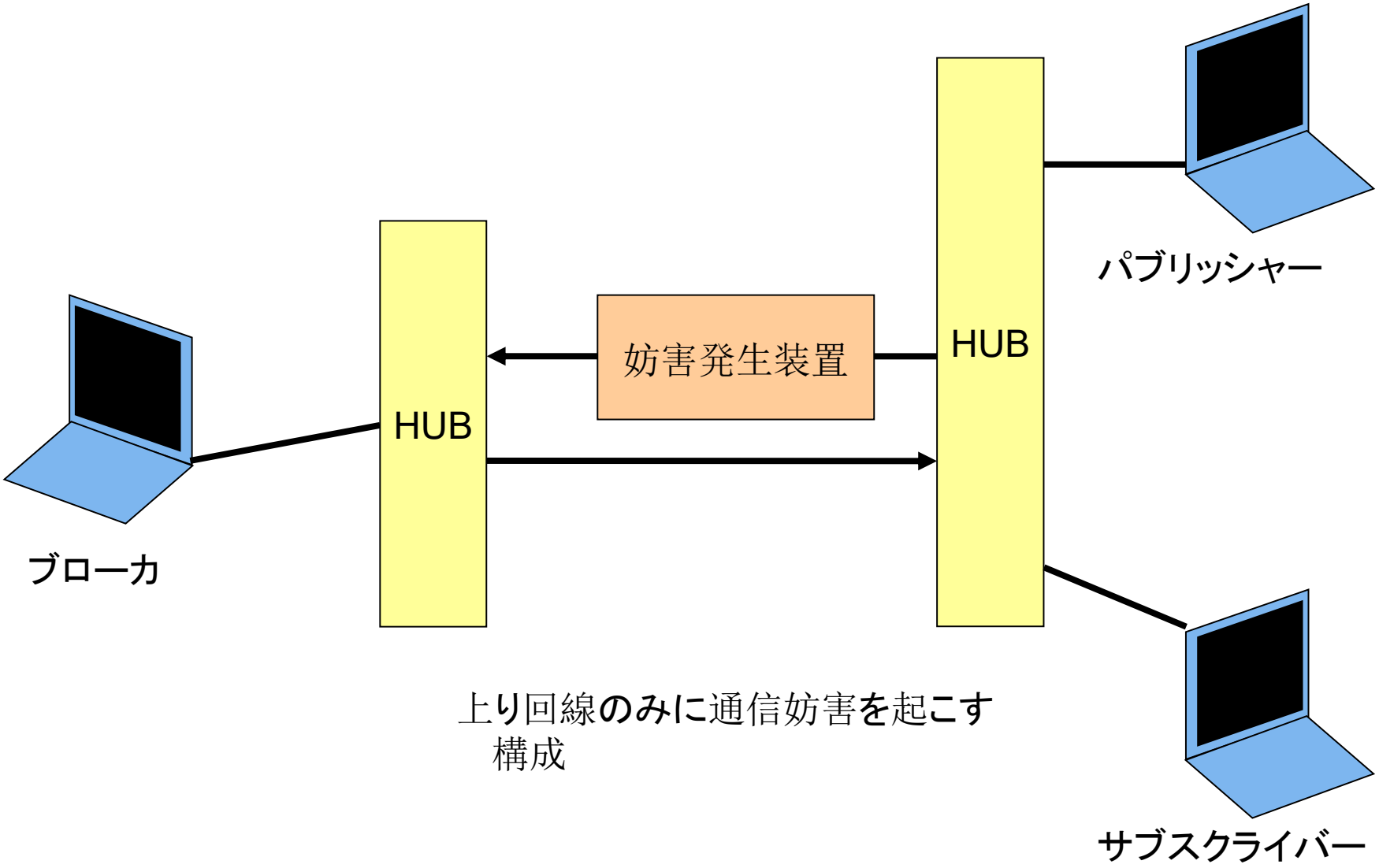
QoS(サービス品質)

QoS 0　メッセージは１回送られる
QoS １　メッセージは最低１回以上届けられる
QoS ２　メッセージ必ず１回届けられる

QoSの測定方法

Publisherから連番のメッセージを送信し
　Subscriberがそれを受信し抜けあるいは重複
　をカウントする。
クライアントとブローカー間の回線に通信妨害
　発生器を設置する。
通信妨害の階級を段階的に上げていき測定する。
通信妨害の階級はpingの損失レートを基準とる。

QoS実験システム構成



上り回線のみに通信妨害を起こす構成

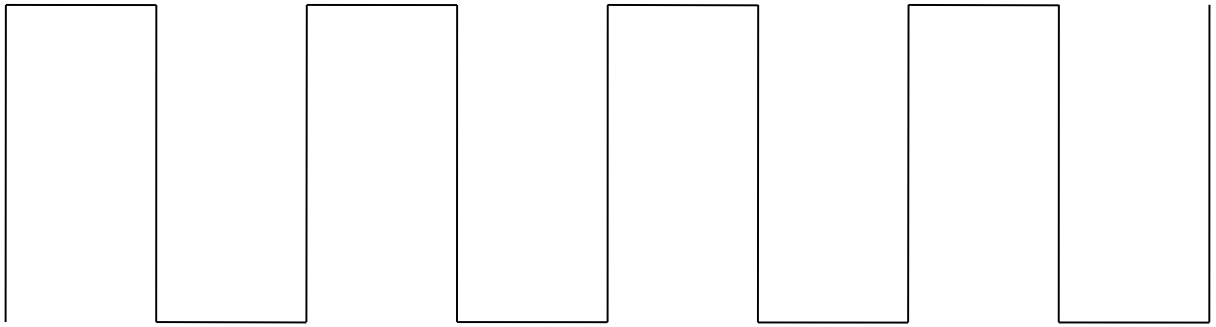
通信妨害発生プログラム（一部抜粋）

```
while(1) {
  digitalWrite(13, HIGH);
  delay(500);                    // 回線がショート状態の時間(ms)
  pinMode(13,LOW);
  delay(500);                    //回線が通常状態の時間(ms)
}
```

回線のショート状態と通常状態を一定間隔で繰り返す。

回線がショート状態

回線が通常状態



C:\Users\michihiro\Dropbox\SC\SC2017\NoiseGen\noiseGenerator

Pub/Sub 1 対 1 の場合

障害の周波数を変える（但しデューティ比は50%）

PUB	SUB	損失	重複	ping損失	time
QoS 0	QoS 0	0	0	0	6
QoS 1	QoS 1	0	0	0	8
QoS 2	QoS 2	0	0	0	6
QoS 0	QoS 0	0	0	65% ※1	92
QoS 1	QoS 1	0	0	65% ※1	88
QoS 2	QoS 2	0	0	65% ※1	73
QoS 0	QoS 0	11	0	63% ※2	267
QoS 1	QoS 1	9	0	63% ※2	258
QoS 2	QoS 2	12	0	63% ※2	301
QoS 0	QoS 0	31	0	69% ※3	800
QoS 1	QoS 1	24	0	69% ※3	772
QoS 2	QoS 2	17	0	69% ※3	798

※1　500ms間隔で回線がショート状態となる。　※2　250ms ※3　100mS

考察

通信妨害の頻度（周波数）を変えても
　ping(UDP)にはそれほど影響なし。
MQTTなどTCP系のプロトコルでは、回線が
　つながっているトータルの時間は同じでも(50%は回線がつながっている)障害の
　頻度が増すと、エラー率およびスループットに著しい悪影響を及ぼす。
通信妨害の程度が大きいとQoS2の方がエラー率において多少有利。

独自プロトコルとの比較

UDPであるICMPが障害の程度に関わりなくそれなりにパケットが通ることからUDPベースのプロトコルを試してみたい。

通常のシーケンスでは回線がショートする度にARPのやり取りが発生するが、ターゲットのMACアドレスは固定とする。

まとめ

ご清聴ありがとうございました。

今岡　通博