

MQTT プロトコルを用いた WIN システム用 リアルタイム波形表示システムの開発

鶴 岡 弘^{*†}

Development of Real-time Monitoring System for WIN using MQTT Protocol

Hiroshi TSURUOKA^{*†}

Abstract

A real-time monitoring system applicable to the WIN system was developed by combining Mosquitto, which implements the MQTT protocol, smoothie.js, a JavaScript library that enables easy real-time plot, and a converter that converts WIN-format data to JSON-format data. This system simplifies checking waveform data when installing a new station.

Key words : WIN system, MQTT, JavaScript, Real-time monitor

はじめに

WIN システム (東京大学地震研究所地震火山情報センターウェブサイト) は, 多チャンネルの地震波形データを取り扱うための処理システムで, UNIX 上で動作する多くのプログラム群から構成されている. また, 同システムは, 地震観測テレメータシステムにおいて, その波形フォーマットを含め, リアルタイムデータ交換のためのシステムとして広く利用されている. プログラム群の中には, データ交換の際に利用される共有メモリ上のデータをテキストで表示するプログラム `shmdump` があり, これと連携して, ダンプされたテキストから時系列を図示するための `Tcl/Tk` で作成された波形表示プログラム群 (`shmx`, `shmz`, `shmck`) がある (図 1 (鶴岡, 2003)). 波形表示プログラムは, `Tcl/Tk` で記述されているためマルチプラットフォーム対応となっているがスマートフォン等へはまだ未対応である. 観測点新設時における波形データのチェックなどにおいては, スマートフォンなどの端末で実施できると非常に効率がよく, さらに Web ブラウザでのアクセスが可能であればより望ましい. ただ, Web ブラウザによるリアルタイム表示システムは存在しているものの, 1 秒ごとに波

形表示用のファイルを作成して, それらを JavaScript のプロットライブラリによって表示するシステムなどが開発されている程度である (伊藤, 2018). その不足を改善するため, Machine-to-Machine (M2M) や Internet of Things (IoT) においてよく利用されている Message Queueing Telemetry Transport (MQTT) プロトコルを用いた WIN システム用リアルタイム波形表示システムを開発したので報告する.

MQTT とは

MQTT は, 機械同士が通信を介して情報をやり取りする M2M や家電, 自動車など多種多様なモノがインターネットにつながりお互いに情報をやりとりする IoT においてよく利用されているプロトコルであり, シンプル, 軽量, 省電力という特徴を持っている. 今回はその実装アプリである Mosquitto (Eclipse Mosquitto ウェブサイト) を用いてシステムの開発を行なった. MQTT は, パブリッシュ/サブスクライブ型モデル (図 2) を採用しており, 1 対 1 はもとより多対多のやり取りが可能な応用範囲の広いプロトコルである.

リアルタイム波形モニタリング

MQTT プロトコルを用いて地震データのやり取りを行い, リアルタイムの波形表示については, JavaScript ライブラリの `smoothie.js` (Smoothie Charts ウェブサイト) を

2022 年 11 月 1 日受付, 2022 年 12 月 23 日受理.

[†] tsuru@eri.u-tokyo.ac.jp

^{*} 東京大学地震研究所地震火山情報センター

^{*} Earthquake and Volcano Information Center, Earthquake Research Institute, the University of Tokyo

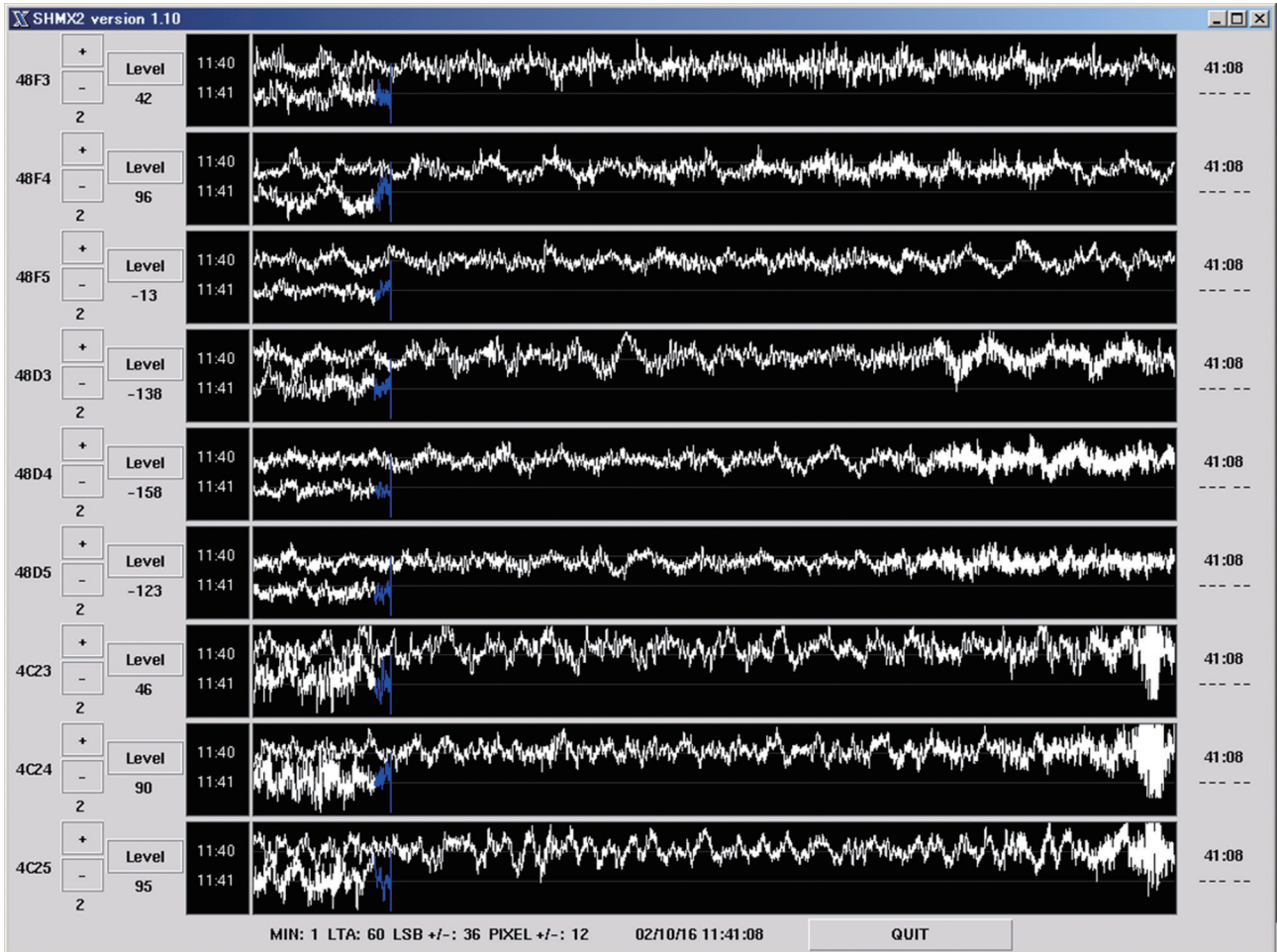


図 1. shmdump+shmx によるリアルタイム波形表示. 青色部分の波形は最新の波形データのプロットである.

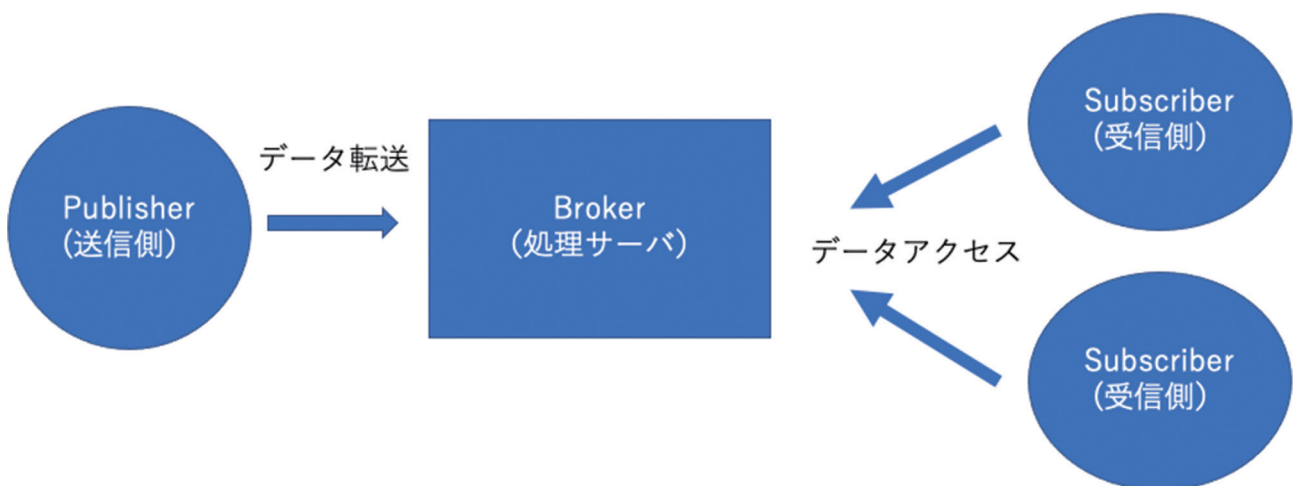


図 2. MQTT のモデル

採用した. さらに, Mosquitto broker への配信データフォーマットとしては, WIN のテキスト形式 (WIN アスキーフォーマット) データを JSON 形式 (WIN JSON フォー

マット) データに変換して送信することとした. WIN アスキーフォーマットおよび WIN JSON フォーマットの例は表 1 に示した. JSON 形式への変換を採用したのは,

表 1. WIN アスキーフォーマットおよび WIN JSON フォーマットのサンプル

<pre>WIN アスキーフォーマット 21 03 17 13 04 22 1 0252 5 267 308 249 169 7 WIN JSON フォーマット {"t":["21,3,17,13,4,58"],"n":1,"ch0252":{"f":5,"d":["54,229,393,545,510"],"chs":["0252"]}</pre>

表 2. WIN アスキーフォーマットから WIN JSON フォーマットへの変換プログラムソース

```
#!/usr/bin/perl
$|=1;
while (<STDIN>) {
    $line = $_;
    @X = split(' ', $line, 9999);
    $yr = $X[0]; $mo = $X[1]; $dy = $X[2]; $hr = $X[3]; $mi = $X[4]; $sc = $X[5]; $nch = $X[6];
    $yr = sprintf("%d",$yr); $mo = sprintf("%d",$mo); $dy = sprintf("%d",$dy);
    $hr = sprintf("%d",$hr); $mi = sprintf("%d",$mi); $sc = sprintf("%d",$sc);
    $nch = sprintf("%d",$nch);
    printf("%Y{Y}tY":Y[$yr,$mo,$dy,$hr,$mi,$scY],Y"nY":$nch,");
    $N1=$nch-1;
    $CHS="YchsY":Y[";
    for ($i=0;$i<$nch;$i++) {
        $line = <STDIN>;
        @Y = split(' ', $line, 9999); $data{$Y[0]}=$Y[2];
        printf("YchY[0]Y":Y{YfY":$Y[1],YdY":Y[";
        $N=$Y[1]-1; $CHS = $CHS . "Y"Y[0]Y",";
        if ( $i == $nch -1 ) {
            chop($CHS); $CHS = $CHS . "Y]";
        }
        for($j=0;$j<$Y[1];$j++){
            if ( $j < $N ) { printf("%d,", $Y[2+$j]); } else { printf("%dY]Y)", $Y[2+$j]); }
        }
        if ($i<$N1) { printf(","); } else {}
    }
    printf(", $CHSY}Yn");
}
}
```

smoothie.js を用いた描画において、実装が容易になるためである。また、XML などのマークアップ言語での変換も考えられたが、データ表現としては非常に汎用性が高いものの、データ量が大きくなりがちになるため、コンパクトな仕様かつ最近広く利用されるようになったデータ交換フォーマットである JSON を用いた。表 2 に WIN アスキーフォーマットから WIN JSON フォーマットへ変換する Perl のプログラムを示した。

実 際 例

実際に開発されたシステムのスナップショットを図 3 に示した。図 3 では、筑波観測点に設置された STS-1 地震

計の 3 成分をプロットしている。ブラウザの負荷は非常に軽く、スマートフォンなどの比較的非力な端末のブラウザでも WIN データの 1 秒パケット分を遅延なく十分高速に表示できることを確認した (図 4)。また、CPU 負荷の比較として、地震研究所の Youtube チャンネルにあるコンテンツ「30 秒でわかる地震研」(Earthquake Research Institute, Univ. Tokyo Youtube チャンネル, 2021) を再生した際の最大の負荷と筑波観測点の広帯域地震計 3 成分 1 分間表示の負荷は平均して 10 対 1 であった。この結果は Youtube の動画を再生できれば問題なく波形モニターができることを意味している。

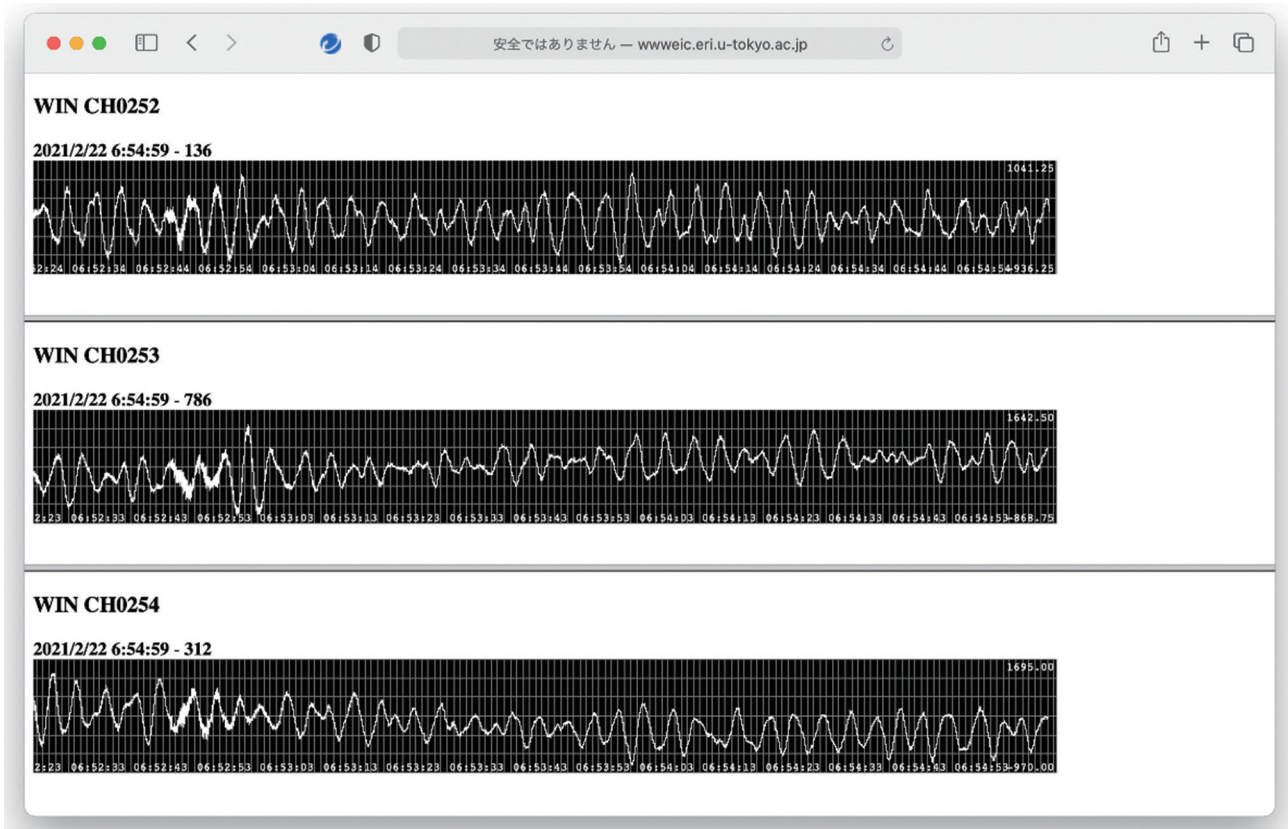


図 3. 実際の表示例

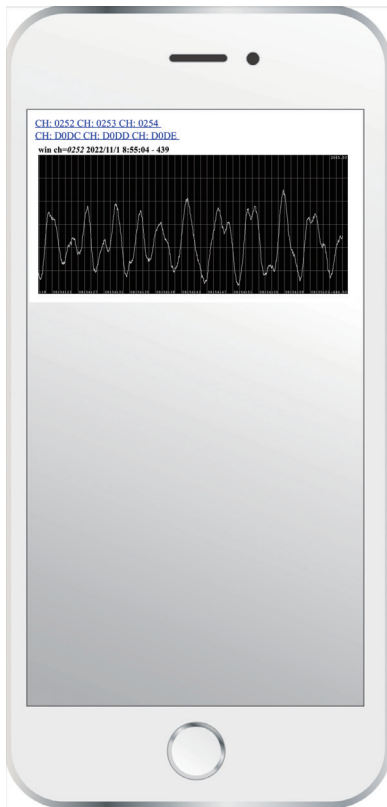


図 4. スマートフォンでの表示例（地震波形の画像はスマートフォンでのスクリーンショットであり、イメージ図である）

結論および今後の展望

既存のツールであり、かつ容易に入手可能な Mosquitto および smoothie.js を活用して WIN システム用のリアルタイム波形表示システムを開発した。今回は、Mosquitto を利用したために、データ伝送はアスキー形式のみの対応となっているが、WIN パケット（バイナリ）を送信できるとデータ転送サイズ等の面からも効率的であり、今後はその方針で開発を進める予定である。また、MQTT プロトコルを採用すればデータ送受信にアクセス制限をかけることなどが可能となり、より柔軟なデータ送受信を実現できる。さらに、MQTT プロトコルは WIN フォーマットのデータ送受信に使われているプログラムである recvt/send_raw に代わって利用できる可能性があり、その性能評価なども併せて実施していく予定である。

謝 辞：技術研究報告編集委員会と 2 名の匿名査読者の方々には、有益なご指摘を頂きました。ここに記して感謝申し上げます。

文 献

Earthquake Research Institute, Univ. Tokyo Youtube チャンネル, 2021, 30 秒でわかる地震研, <https://www.youtube.com/>

- watch?v=trEdTaeBSmg, (参照 2022-11-01).
- Eclipse Mosquitto ウェブサイト, Mosquitto, <https://mosquitto.org>, (参照 2022-11-01).
- 伊藤貴盛, 2018, JavaScript による地震波形表示の試み, 日本地球惑星科学連合 2018 年大会, STT50-P09.
- Smoothie Charts ウェブサイト, smoothie.js, <http://smoothiecharts.org>, (参照 2022-11-01).
- 東京大学地震研究所地震火山情報センターウェブサイト, WIN システムとは?, <https://www.eri.u-tokyo.ac.jp/WIN/index.html>, (参照 2022-11-01).
- 鶴岡 弘, 2003, WIN システム用波形モニターツールの活用, 震研技報, **9**, 14-19, doi : 10.15083/00032207.