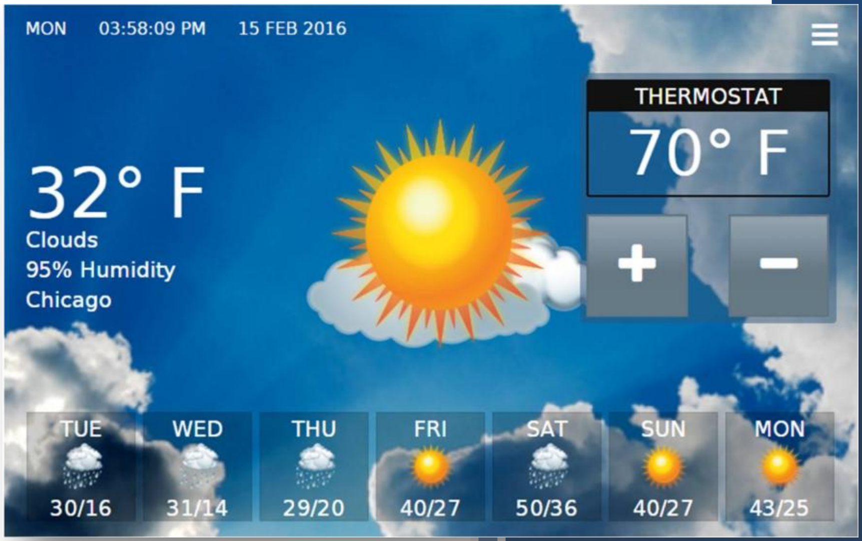
2016

SMQ Weather



Real Time Logic LLC

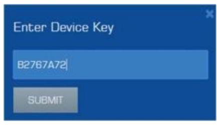
INTRODUCTION

The SMQ Weather Application illustrates an example of the SMQ Architecture capability as used in web-based control of an IoT Device Thermostat. The example provides cloud-based dynamic control of the client via on-demand content generation and settings manipulation from either a local display or a remote browser-based HMI.

The following instructions are for using a simulated device HMI running as a DOS command application. If you have the SMQ Weather Application in a device, you may substitute the simulated device with the real device.

DESKTOP (SETUP & USE)

1. Download and start simulated device: https://realtimelogic.com/downloads/SmqWeatherApp.exe
2. Make note of the eight digit Device ID: **>>> Device ID: B2767A72 <<<**
3. Using a Web-Browser navigate to: https://simplemq.com/WeatherApp/
4. Select the Mobile Menu Icon: ≡
5. Select Connect: Connect
6. Submit the eight digit Device ID provided by the client: **B2767A72**

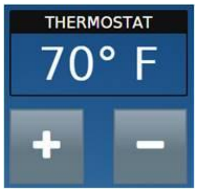


Enter Device Key

B2767A72

SUBMIT

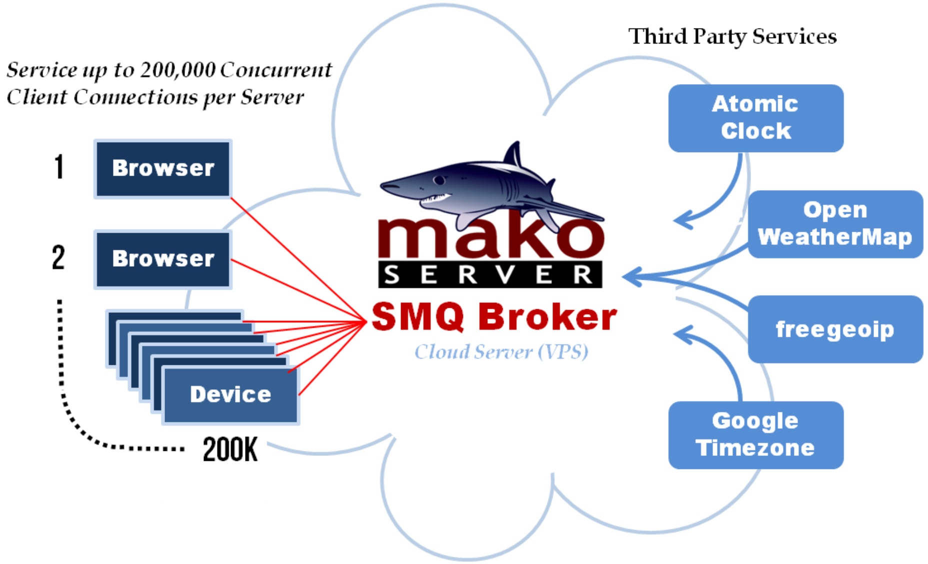
7. The Web-Browser will automatically connect and synchronize to share data and engage Thermostat control upon entering the Device ID.



8. Adjust temperature settings by selecting **Up (+)** or **Down (-)**.
9. Open a second Web-Browser and repeat above Steps 3-6.
10. Change City Information by selecting the Mobile Menu Icon ≡ and navigating to: Set City
11. Notice how new weather & forecast data is synchronized between all connections.
12. To adjust Celsius to Fahrenheit settings toggle Set Celsius for personal preference.
13. The About Menu About provides additional information about the SMQ Weather Application.

DESCRIPTION

The following figure illustrates how client connections (device or browser) receive data from third party services via the SMQ Broker, which is located on the cloud server:



DEVICE

The device first connects to the cloud server using a hard coded URL compiled into the device code. The URL can be changed by anyone with access to the SMQ Weather Application device Source Code. After connection occurs, the display (UI) is updated with the current regional weather and forecast information based on the geo-location of the simulated device. The Connect Menu within the UI requires a unique device ID, which is generated by the SMQ client and displayed in the device console window. The (number) is used as the device credentials and makes it possible for a browser to connect and communicate with the unique device.

The server sends the information provided by the third party *(Atomic Clock)* service to the device. The GMT time provided by the atomic clock is then adjusted according to the device geo-location or local time zone for the device. The *(Google Timezone)* service provides the zone offset based on geo-location. Once the device establishes a base time, an internal counter is started updating base time by one every second. This time may drift, but the server will send *(Atomic Clock)* updates to the device approximately every 15 minutes.

BROWSER

Upon first connection to the online server, the SMQ Weather Application is immediately loaded and the related browser based geo-location weather information is displayed within the session window. Selecting (Connect) from the drop down menu and entering a Device ID will establish a (device to browser) communication via the SMQ Weather Application server code. The browser and device sync and the server sends a new update to the display refreshing it with Device based geo-location weather and forecast information. Entering the Device ID also enables a Thermostat overlay for temperature manipulation via the session window.

THIRD PARTY SERVICES

- freegeoip.net: IP address to geo-location translation
- openweathermap.org: Weather and forecast information by geo-location
- Google Time Zone API: Provides time zone offset based on geo-location.
- time.nist.gov: Atomic Clock, makes the time in the device accurate.

ABOUT SMQ

SMQ, (Simple Message Queues), is an easy to use IoT/ M2M publish subscribe protocol designed and optimized for embedded systems providing instantaneous Edge Node connectivity. The solution allows resource constrained real-time control, analysis, and updates.

MESSAGE STRUCTURE, is Pub/Sub pattern based operation for propagating data close to real-time creating a practical environment, while taking advantage of Web 2.0 usage by Restful web services. Messages are coordinated and integrated among software components as software applications change over time, and are transferred from one machine to another over unreliable wireless networks.

FEATURES

- Publish-Subscribe
- Asynchronous Communication
- 1-to-1 Messaging via P2P Tunnel
- 1-to-Many Messaging
- Ephemeral Topic IDs
- RPC Compatible
- Hierarchical Topic Structures
- Client Supervision
- Designed for Constrained Devices
- Persistent Messages
- Web of Things IoT Bridging
- Light on the Network