

Use MQTT to Send Data from Particle Photon to IBM Watson IoT Platform – A C++ Sample Code

Dr. Xiaohai Li (xhli@citytech.cuny.edu)

Robotics & Intelligent System Research Lab
Dept. of Computer Engineering Technology
New York City College of Technology

Spring 2018

The sample code here is written for Particle Photon to send data to IBM Watson IoT Platform via MQTT. After successfully sending your data to IBM Watson IoT Platform, you can use the visualization tools (“Boards” and “Cards”) available on Watson IoT Platform to visualize the data.

The MQTT library used in the code is accessible from Particle Build, which was written by Hirotaka Niisato [1]. Thoroughly read the following code and understand how MQTT works, ie., how to use Niisato’s library to send data to the cloud. Most importantly, know how to modify this code for sending your own data.

Note: Due to possible update of the MQTT library, the code here might need some modification. Search and read MQTT.h and MQTT.cpp on Particle Build or Niisato’s GitHub repository[1] for complete details of the latest MQTT library.

To help you understand the code, some of the important comments are highlighted.

	This code is to send data to IBM Watson IoT platform from a variable named as “testdata” (defined on line32 & 58).
1	/* Author: Xiaohai Li
2	* License: GPL v2.0
3	* 04/05/2018
4	*/
5	
6	// This #include statement was automatically added by the Particle IDE.
7	#include <MQTT.h> //search and include the mqtt library via Particle online IDE
8	
9	#define HOST_PORT 1883 //port# used by host (IBM Watson IoT Platform): 1883 (default)
10	//change part# to 443 for secure connection
11	
12	#define MQTT_QoS 0 //MQTT QoS = 0: message will be delivered zero or once (default)
13	//MQTT QoS = 1: message will be delivered at least once
14	//MQTT QoS = 2: message will be delivered exactly once
15	
16	//Use your device information to fill in the following code in line17~21
17	char *MQTT_HOST = "your_organizationID_here.messaging.internetofthings.ibmcloud.com";
18	char *MQTT_CLIENT = "d:your_organizationID_here:your_Device_Type_here:your_DeviceID_here";
19	//an example of MQTT_CLIENT is "d:myorgID:photons:photon_a1";
20	char *MQTT_USERNAME = "use-token-auth";
21	char *MQTT_PASSWORD = "your_token_here";
22	
23	// MQTT Message Topic of the data that will be sent to Watson.
24	// iot-2 --> The protocol
25	// evt --> Specifies the message type, use "cmd" for applications
26	// testdata --> Name of message //it is also Event's name. Important! Needed by cloud!
27	// fmt/json --> Message will be send in JSON format // Watson use JSON format as default!
28	char *MESSAGE_TOPIC = "iot-2/evt/testdata/fmt/json"; // Segment can be customized by you
29	
30	MQTT client(MQTT_HOST, HOST_PORT, callback); //Create a MQTT client for your device
31	
32	int testdata = 0;
33	int IndicatorLed = D7;
34	
35	char payload[80]; //MQTT message's payload. A string in JASON format.
36	// 80Bytes used here. Change the size according to your data.
37	
38	
39	void setup() {

```
40 pinMode(IndicatorLed, OUTPUT);
41
42 Serial.begin( 9600 ); //Use serial monitor to debug the code.
43 //Read "Particle serial tutorial" for detailed how-to.
44 Serial.println( "Connecting Photon to IBM Watson IoT Platform ..... " );
45
46 while( !Serial.available() ) {
47     Particle.process();
48 }
49
50
51 client.connect(MQTT_CLIENT, MQTT_USERNAME, MQTT_PASSWORD); //Connect Photon to Watson IoT Platform
52
53 if( client.isConnected() ) { //Verify the connection
54     Serial.println( "Now connected!" );
55     // client.subscribe( MQTT_SUBSCRIBE ); //This sample code is only to send data to cloud,
56     //not subscribe from cloud, so this line is commented out.
57 }
58 }
59
60 void loop() {
61     testdata ++;
62     if (testdata>500) testdata = 0;
63
64     //Convert data to a JSON format string on line66:
65     //JSON format example: {"property": value1, "property2": value2, ...}
66     sprintf(payload, "{ \"testdata\": \"%d\" }", testdata);
67
68     //send the JSON formatted payload (including the data) to Watson under the pre-defined topic
69     client.publish(MESSAGE_TOPIC, const_cast<char*>(payload) );
70
71     digitalWrite(IndicatorLed, HIGH); //Quickly blink the LED D7 once when a data is sent
72     delay(100);
73     digitalWrite(IndicatorLed, LOW);
74     delay(100);
75
76     client.loop();
77
78     Serial.print( "Data being sent to Cloud: " ); //Display the sent data on SerialMonitor
79     Serial.println(testdata);
80
81     delay( 3000 ); //Set the period of sending data
82 }
83
84 void callback( char* topic, byte* payload, unsigned int length ) {
85     char p[length + 1];
86
87     memcpy( p, payload, length );
88     p[length] = NULL;
89
90     String message( p );
91 }
```

Note: Refer to Particle Serial Tutorial to learn how to use Serial Monitor with Particle for testing and debugging.

Link: <https://community.particle.io/t/serial-tutorial/26946>

or: https://github.com/rickkas7/serial_tutorial

References

1. <https://github.com/hirotakaster/MQTT>
2. Kevin Hoyt, "Particle Photon on Watson IoT", April 27, 2016, retrieved on February 10, 2018 at URL: <https://www.kevinhoyt.com/2016/04/27/particle-photon-on-watson-iot/>
3. <https://github.com/ibm-watson-iot/iot-cpp>