Explore an architecture for connected assets in Maximo

Connect sensors to a NodeMCU board, and use Watson IoT Platform to receive and send data over to Maximo Asset Health Insights

Louis Stoop

January 18, 2017

Learn how to implement an asset management architecture by using inexpensive sensors, which are connected to Watson IoT Platform, which is then linked to Maximo Asset Health Insights.

Many Maximo customers have SCADA/DCS systems in place for their most critical assets. However, they also have many other potential measurement points in their manufacturing facility, which they would like to measure, but these other assets do not justify the use of a SCADA/DCS system, as the costs would not be worth it.

To learn more about Maximo Asset Health Insights, you can watch this video.

In this tutorial, we show you how to set up a connected manufacturing and asset management architecture for these other assets. You'll use cheap sensors connected to a NodeMCU board to take measurements and send them to the IBM Cloud offering with Watson IoT Platform. Then, you'll feed that data into Maximo, specifically Maximo Asset Health Insights, which is an add-on product that runs on top of the Maximo Workcenter that is part of Maximo V7.6.0.5 or later. Then, from Maximo, you will be able to read those measurements and store aggregated data in your Maximo Meters, and also view monthly historical charts in your Maximo dashboard.

You'll need to work with your Maximo Administrator to complete the second half of this tutorial.

If you are convinced of this architecture, you can swap out and use more rugged sensors.

Let's get started!

What you’ll need for this IoT app

Like with all electronics, be careful when you connect any sensors or boards to each other and to the power supply. Mixing up positive and negative wires do result in your components...
You'll need the following devices or sensors:

- An NodeMCU board. The NodeMCU board is one of the cheapest and easiest to use because it uses the Arduino architecture to program the board. The current version is V3, which is what I use in our sample app. Check your local electronics store, or use AliExpress, which offers cheap prices, but it might take a while (4 - 6 weeks) to receive your board. This board has multiple digital inputs and only one analog input. It can run on 5 V or 3.3 V power. The microUSB port can serve as your power supply.

- A temperature sensor, such as the DHT22 temperature sensor that I use in our sample app. This sensor is accurate to 0.1 degrees Celsius. There is a slightly cheaper option, the DHT11, but that one is accurate to only 1 degrees Celsius, so I would not recommend that sensor. You can use many types of sensors, such as light sensitivity sensors, motion detection sensors, or vibration detection sensors. You can use any type of sensor that you like (as long as it's compatible with the analog or digital ports on the NodeMCU board) and still use the code in this tutorial. However, you'll need to modify the code to fit your sensor type.

- Optionally, a breadboard or baseboard for your NodeMCU board. This board will give you the option to have more pins to play around with (if you get more sensors), and the option to add a 12 V power source to keep your board safe. If you only plan to use the microUSB as a power source, you do not need this board.

In addition to the devices and sensors, you also need:

- An IBM Cloud account. (You can request a free trial here.)
• Familiarity with IBM Cloud Platform. You need to know how to browse the catalog and instantiate one of the services. (You can complete the tutorials in the IBM Cloud essentials learning path to gain a complete understanding of IBM Cloud.)
• Maximo 7.6.0.6 or higher. Also, Maximo Asset Health Insights 7.6.0.1, which is an add-on to Maximo and is licensed separately.

In the second part of this tutorial, you'll need to work with your Maximo Administrator to configure it to receive and save the sensor data.

1. Installing and configuring drivers for connecting to the NodeMCU device

The NodeMCU uses specific COM port drivers, which you need to install.

Do not connect your NodeMCU to the serial bus until you have installed these drivers. (If you already connected the NodeMCU board, you will have to disconnect and reconnect it after you install the driver so that the board will be recognized.)

1. You can find the latest drivers by searching on "ch340g nodemcu drivers," or you can find them in this nodemcu-devkit GitHub repo:
   https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers
2. Install the CH341 serial port drivers.

2. Installing and configuring the Arduino software and libraries

Because every board and sensor has their own specific characteristics, we will need to load the correct libraries for the Arduino software.

1. Download and install the Arduino software from this site:
   https://www.arduino.cc/en/Main/Software
2. Open the Arduino software.
3. Click Files > Preferences.
4. In the Additional Boards Manager URL, specify this URL:

   Then, click OK to close the Preferences window.
5. To load the specific board type for our NodeMCU device, select Tools > Board: [your board] > Boards Manager.
6. After a few seconds, the Boards Manager window opens with the boards from the Internet and the specific location we just specified. In the Search field, enter “ESP8266” which is the CPU chip that is used in the NodeMCU device.

7. In the search results, select the ESP8266 entry to install the board.

8. If prompted, restart your Arduino software.

9. To select our NodeMCU board as the default board, select **Tools > Boards > NodeMCU V1.0**. When you select the correct board, the Arduino software can communicate with the NodeMCU board when you load programs on to it. If you select the wrong board, you'll get errors when you compile the code (unknown definitions) and you'll want to double-check the board that you selected as your default board.

### 3. Loading libraries for the devices and sensors

Every type of sensor might have specific libraries. For example, digital sensors often have a data pin, that sends encoded data that needs to be interpreted by a library. For our sample app, we need to load the following libraries:

- ESP8266 Library, for working with our NodeMCU board (verify that this library is already loaded for our board)
- DHT Sensor Library, for working with our temperature sensor
- PubSubClient library, for communicating with the IBM Cloud platform

To load the libraries, complete these steps for each of the above libraries:

1. In the Arduino software, select **Sketch > Include Library > Manage Libraries**.
2. In the search bar, search for the library.
3. In the search results, select the library to install.

### 4. Loading Watson IoT Platform code into your Arduino software

We are now ready to bring in the first code to your Arduino software. Do not execute the program yet, as we will need to modify a number of parameters, depending on your IoT settings and your wifi settings.

1. In the Arduino software, create a new sketch. A sketch is a program or unit of code that is uploaded and run on an Arduino board.
2. Copy and paste the following lines of code into this sketch.

```cpp
/** Sensor for Maximo IoT Nov 2016
 * Author: IBM */
```
* License: Apache License v2

```c
#include <ESP8266WiFi.h>
#include <PubSubClient.h> // https://github.com/knolleary/pubsubclient/releases/tag/v2.3
#include <DHT.h>
#define DHT11_PIN D1
#define DHTTYPE DHT22

//-------- Customize these values -----------
const char* ssid = "YOUR SSID";
const char* password = "YOUR PASSWORD";
#define ORG "YOUR ORGID"
#define DEVICE_TYPE "YOUR DEVICE TYPE"
#define DEVICE_ID "YOUR DEVICE ID"
#define TOKEN "YOUR TOKEN"

//-------- Customize the above values --------
DHT dht(DHT11_PIN, DHTTYPE);
char server[] = ORG "\".messaging.internetofthings.ibmcloud.com";
char topic[] = "iot-2/evt/status/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;

PubSubClient client(server, 1883, NULL, wifiClient);
void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.print("Connecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println(" ");
  Serial.println("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}
void loop() {
  if (client.connected()) {
    Serial.println("Reconnecting client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      Serial.println(" ");
    }
  } else {
    Serial.println("Publish failed");
  }
  delay(20000);
}
```

// in the string below, where you see quotation marks followed by a backslash, this means the quotation marks should be included in the final payload.

String payload = "{"d":{"SensorXX":"Usage"; //Name this anything you like. Just make sure to keep the quotation marks in place.
payload += ","Temperature":"; payload += dht.readTemperature();
payload += ","Humidity":"; payload += dht.readHumidity();
payload += "}"};
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(topic, (char*) payload.c_str())) {
  Serial.println("Publish ok");
} else {
  Serial.println("Publish failed");
} delay(20000);
```
3. Modify the values in lines 12 - 13 to configure your wifi network. For SSID and password, specify the credentials for your wireless network. Later in this tutorial, we will configure the Watson IoT Platform values in lines 14 - 17 (ORG, device_type, device_ID, and token).

This sketch only sends values for the temperature and humidity from our DHT22 sensor. You can use this code as an example for how you might extend this sketch to add more sensors, such as light sensors or flow sensors.

5. Step 5: Connecting your DHT22 Sensor to your NodeMCU board

In a generic Arduino sketch, the assigned pin layout is always referred to with a specific number. On the NodeMCU, you can also use the D1 designation (for Digital 1 for example) to be more specific. Pin 1 in a generic sketch for nodeMCU is NOT the same as D1. (Otherwise it would be too easy!) So, always use the Dx pin assignment naming convention and you won't go wrong. Not all Arduino-compatible boards use this method, but for the NodeMCU series it works. If you use any other type of board with a different chip, you might need to research the documentation for that board to understand their pin layout and how that relates to the pin assignment in the sketch.

1. On line 10 of the sketch, there is the definition of the data pin where the DHT22 sensor connects to. Choose either D1 or D2 (not all D pins can be used because some have shared functions).

2. The DHT22 sensor has three pins. Because different sensors have different pin layouts, you need to be careful when you connect the sensor to the board. If you mix up the connections, you can damage the sensor or your board, or both (so much for the 6 USD). Connect the PLUS pin to the 3.3 V pin on the NodeMCU board. Then, connect the MINUS pin to the GND on the NodeMCU board.

3. If you also bought a breadboard or baseboard, make sure that you place the NodeMCU board in the right direction, aligning the wifi symbol on the baseboard with the wifi on the NodeMCU
board. Then, you can use the same pins as in the previous step, but on the baseboard instead.

Later in this tutorial, we will power the board with a microUSB cable. If you choose to use a separate power supply, you can connect any 5 V power to the Vin pin or any 3.3 V power to the 3.3 V input near the microUSB port. If you are using the baseboard or breadboard, it can take a 12 V power supply.

6. Setting up your Watson IoT Platform app in IBM Cloud

Now that you've got the devices and sensors set up, you need to create and configure an IoT app in IBM Cloud. You'll use the Internet of Things Platform Starter boilerplate, which includes the Watson IoT Platform service and a virtual database (Cloudant). You'll need to configure the historical data settings of the Cloudant database for connecting with Maximo data. You'll also need to generate an API key for Maximo which your sensors need to upload their data to the Watson IoT Platform in the cloud.

6a. Creating the IoT service app

1. Log in to your IBM Cloud account. If you do not have an account yet, sign up for a free IBM Cloud account.

1. After login, in the upper right of the screen, click Catalog, search for "IoT", and then select "Internet of Things Platform Starter."
   This boilerplate includes the Node-RED Node.js sample application, a Cloudant database, and the Watson IoT Platform service. While we will not use the Node-RED sample app in this tutorial, I encourage you to explore it after you finish this tutorial.
2. In the App name field, specify a unique name for your app. This name is added to the Host name field, which you can keep the same or customize. Click Create at the bottom of the screen.
3. After it creates your environment, wait for the app to be in Running mode. If it seems to be stopped, click **Start** on the far right of the main screen. It looks like a **Play** button.

### 6b. Configuring the Cloudant database

When you created your IoT app by using the Watson IoT Platform Starter boilerplate, a Cloudant database is created for you. We will need these credentials later in this tutorial when you configure Maximo to the end points. Also, we need to configure the historical data storage settings.

1. In IBM Cloud, on the dashboard for your IoT app, select the CloudantNoSQLDB link to your database.
2. To review the credentials for this database, complete these steps:
   a. Select **Service Credentials**.
   b. Click **New Credential**.
   c. Specify a name for these credentials (for example, Maximo AHI credentials), and then click **Add**.
   d. In the Service Credentials table, click **View Credentials**.
   e. Copy these credentials, and save them to use later.
3. To verify that the Historical Data Storage option is available, complete these steps:
   a. In IBM Cloud, on the dashboard for your IoT app, select the **iotf-service** link for your app.
   b. Click **Launch Dashboard**.
c. From the menu on the left, select the icon that represents **Extensions**.

d. On the Extensions window, if you do not see a card for historical data storage, click **Add Extension** in the upper right corner of the window.

4. To configure the Historical Data Storage settings, complete these steps:
   a. On the Extensions window, click **Setup** under Historical Data Storage.
   b. Select your service to associate your Cloudant database with your service.
   c. Set your bucket interval to **MONTH**, set your time zone to your time zone, and leave your database name as default.
      
      **Warning:** If you forget to set your interval to **MONTH**, you will have trouble later connecting with Maximo.
   d. After you complete the steps above, click **Done**.
   e. A new dialog box will open, and you will be required to confirm this link between IoT platform and your Cloudant. Make sure pop-ups are allowed (once) for you to complete this step.

6c. **Connect your devices (sensors) to Watson IoT Platform**

To have your device data sent to Watson IoT Platform, you need to connect your devices to it and generate secure tokens for each device.

   1. While you are still in the Watson IoT Platform Dashboard, from the menu on the left, click **Devices**.
2. To add a new device for your NodeMCU board, click Add Device.

3. Click Create Device Type. You need to create a device type for NodeMCU boards, which you can use for any additional NodeMCU boards that you might add.

4. In the Create Device Type wizard, click Create Device Type. (We will not need to create a device gateway; this is for more advanced use.)
   a. Specify a name (such as NodeMCU) and description of your device type.
   b. It is not necessary to specify template or metadata, so just click Next on those pages.
   c. Click Create to define the device type.

5. Now that you have created the device type, you can add the device by using that device type. Make sure that your device type is selected, and click Next.

6. In the Add Device wizard, complete these steps to add your device:
   a. In the Device Info page, specify a unique name in the Device ID field. If you plan to use more than one sensor, consider numbering them (Sensor01, for example) by using a permanent marker to write their number on the wifi chip of the NodeMCU board.
   b. It is not necessary to specify metadata or security information on the next pages, so click Next.
   c. Click Add to define the device in Watson IoT Platform.
   d. On the Device summary page that is displayed, you must copy and save the following device credentials: Organization ID, device Type, Device ID and Authentication Token. This dialog box is displayed only once. If you close this dialog box, this token cannot be recovered. **Tip:** Create a simple spreadsheet to save the device names and their tokens.

7. Remember the Arduino sketch we created earlier? You can now open your Arduino software and update the DEVICE_TYPE, DEVICE_ID, and TOKEN values with the above information from your credentials.
//-------- Customize these values -----------
const char* ssid = "YOUR SSID"
const char* password = "YOUR PASSWORD"
#define ORG "YOUR ORGID"
#define DEVICE_TYPE "YOUR DEVICE TYPE"
#define DEVICE_ID "YOUR DEVICE ID"
#define TOKEN "YOUR TOKEN"
//-------- Customize the above values --------

For every new sensor board, you need to repeat the steps above to generate a new name and token for each board.

6d. Generating an API for Maximo

For Maximo to connect with Watson IoT Platform, Watson IoT Platform needs to provide an API key. While we are working in the dashboard for Watson IoT Platform, let's generate that API key.

1. From the left menu, select the icon that represents Apps.
2. Click Generate API Key.

   ![Generate API Key](image)

3. Copy and save the API key and authentication token into a spreadsheet. We will use this later when we configure Maximo.
4. Click Generate.

7. Loading the sketch onto the NodeMCU board

Now that your sensors are configured, connected to Watson IoT Platform, and the Arduino sketch is updated with the generated device credentials, you are ready to load the Arduino sketch on to the NodeMCU board.

1. Connect the microUSB cable to the NodeMCU board and to your computer.
2. From the Arduino software, select your NodeMCU board.
3. From the Tools menu, select the correct COM port.
4. To see what's happening on your board, open the Serial Monitor. From the Tools menu, select Serial Monitor.
5. Review the messages in the Serial Monitor. You can also check your sensors in the Watson IoT Platform dashboard, looking for "Publish OK" success messages.

Troubleshooting tips:

- If you changed the Arduino sketch, all text must appear in quotation marks, whereas numbers must not appear in quotation marks. To add quotation marks in a sketch, include a backslash after the quotation marks ("") to include the quote as text in the payload to be sent to the server.
- Make sure that your token, device type, and device ID are correct. These values are case-sensitive.
- If you see any error codes, verify that you added all of the needed libraries. Review the error messages to help give you hints about what might be wrong. For example, errors like "Unknown command DHTxx" points to a missing library.

8. Configuring Maximo and Maximo Asset Health Insights to connect with Watson IoT Platform and our IoT sensors

Now that the devices and sensors are all connected to Watson IoT Platform, you need to work with your Maximo Administrator to configure a connection (or integration) between Watson IoT Platform and Maximo.

1. Because Maximo runs on WebSphere Application Server, you need to configure WebSphere Application Server to trust both Watson IoT Platform and Cloudant. In WebSphere Application Server, you need to generate certificates for each of them. Follow the SSL Configuration information in this blog post and use the credentials from earlier in this tutorial.

2. In Maximo, you need to add the Cloudant credentials (which you saved earlier in this tutorial) to the Maximo endpoint. You can complete this step here at the Maximo End Points, but you can skip it here and configure it later in the Maximo workspace and achieve the same results.

   a. Find the record IOTHISTCLOUD. Open its properties.
   b. For the URL property, Username property, and Password property, paste the text you find between the quotation marks as created earlier in this document.
c. In the Bucketsize field, make sure that "MONTH" is specified. This value controls how often a new database is created by Maximo.

d. For the Database name field, make sure that "default" is the name.

3. Log in to Maximo Workspace. While the traditional Maximo URL would be http://<YourDomainName>/maximo, the URL for the new workspace will be http://<YourDomainName>/maximo-x. This new URL will be used for the Maximo Workspace that is also used by the Maximo Asset Health Insights.

4. Log on to the Workspace environment (maximo-x).

5. In the Configure Integration page, under IoT Connection, click **Test Connection**. Then, click **Next**.

   If your connection fails, check your Internet connection. Also, make sure that you have generated the WebSphere Application Server certificates. Lastly, check the `systemsout.log` file for more detailed error messages.

6. Review the details of your Cloudant configuration (which you specified in a previous step), and then click **Next**.
7. On the screen below, the device types that you configured in Watson IoT Platform are read from your IoT platform. Select the sensors that you want to use in Maximo. For example, we need to select NodeMCU, and then click Register as Asset (or Register as Location). If you got multiple entries, continue to assign any Device Type that you have to an Asset or a Location type meter. When finished, click 'Next'.

8. Finally, you need to link our IoT devices to Maximo meters on the Map Data Schemas screen. For the NodeMCU board, click the pencil icon. Then, click Add Mapping. Then, click Select Meter.
9. In the Add Meter Mapping dialog box, review and update the values for the meter, properties, data collection, and data synchronization. Make sure that your Historian is running at the same interval in the data collection.

There is a section in this screen below for ‘Rules’. This is meant for more advanced users and is not in scope of this document. You can define criteria. For example, only import the data if the value is over 25. See the following link for more technical details. http://bit.ly/2fqV6tg Once completed, click Done.

So, we have defined the translation between what Maximo is calling a meter (for example, "TEMP-C") and the naming convention that you used in your Arduino program (Temperature). In the source code used earlier, we added Temperature and Humidity. So, you will need to repeat the steps 11 and 12 to also register a meter for the Humidity readings in our sketch.

9. Map IoT readings to your assets

So far we have defined a meter type in Maximo to a reading type from the Watson IoT Platform. In the next step, you need to work with your Maximo Administrator to link specific sensors to specific assets in Maximo.
1. From the Menu in the upper right of your Maximo Workcenter screen, click **Associate Devices**.

2. Because you already mapped TEMP-C to Temperature, you only need to link the Sensor ID and Maximo will know what element to use of the sensor data coming in. Use the **Device Type** drop-down list to select the right type to use, such as NodeMCU in our example.

3. Use the **Device ID** drop-down list to select the sensor ID that you created earlier in the sketch. This will uniquely link one sensor from the Watson IoT Platform to one asset in your Maximo installation.

   You will only see meters and assets not yet associated. Once a mapping is made, and you save the mapping, they will disappear from this view but they can still be seen in the Manage Assets tab.

10. **Check your results**

   With your sensors connected through Watson IoT Platform and mapped to assets in Maximo, you can work with your Maximo Administrator to view the meter readings in Maximo.

   If you have old meter readings that you want to delete or update, you can work with your Maximo Administrator to delete timers for cron tasks, update the timers for cron tasks, and re-run cron tasks as appropriate.

1. Wait for the crontask to run. Check for the meter readings with one of these steps:
   a. In the Asset app, click **Manage Meter Readings**.
   b. Open the Maximo Workcenter in your browser ([http://yourdomain/maximo-x](http://yourdomain/maximo-x)). Log in, and select any asset that you added.
   c. Select **Meter Readings**.

2. When you initially open the meter readings for the asset, you will only see the green dots, which represent the Maximo measurement points.
Within 10 - 15 seconds (or so, depending on network speed), Maximo retrieves a longer history (up to a month) from the Cloudant database and plots those values in the same chart.

3. Zoom into your views by clicking and holding anywhere on the chart, dragging your cursor sideways, and releasing the mouse button.

If you zoom horizontally and vertically, you can also zoom into a specific section of your chart.

Conclusion

In this tutorial, you learned how to configure a sensor, link your sensor data to the Watson IoT Platform, and then link it into Maximo that is running Maximo Asset Health Insights. Do you have ideas on how you can implement this use case in your environment? If you have any thoughts, questions, or comments, please post them on this article.
Related topics

- Watson IoT Platform documentation
- Maximo Asset Health Insights documentation
- Maximo Asset Management documentation