

# Delivering Arduino Data to Thingworx

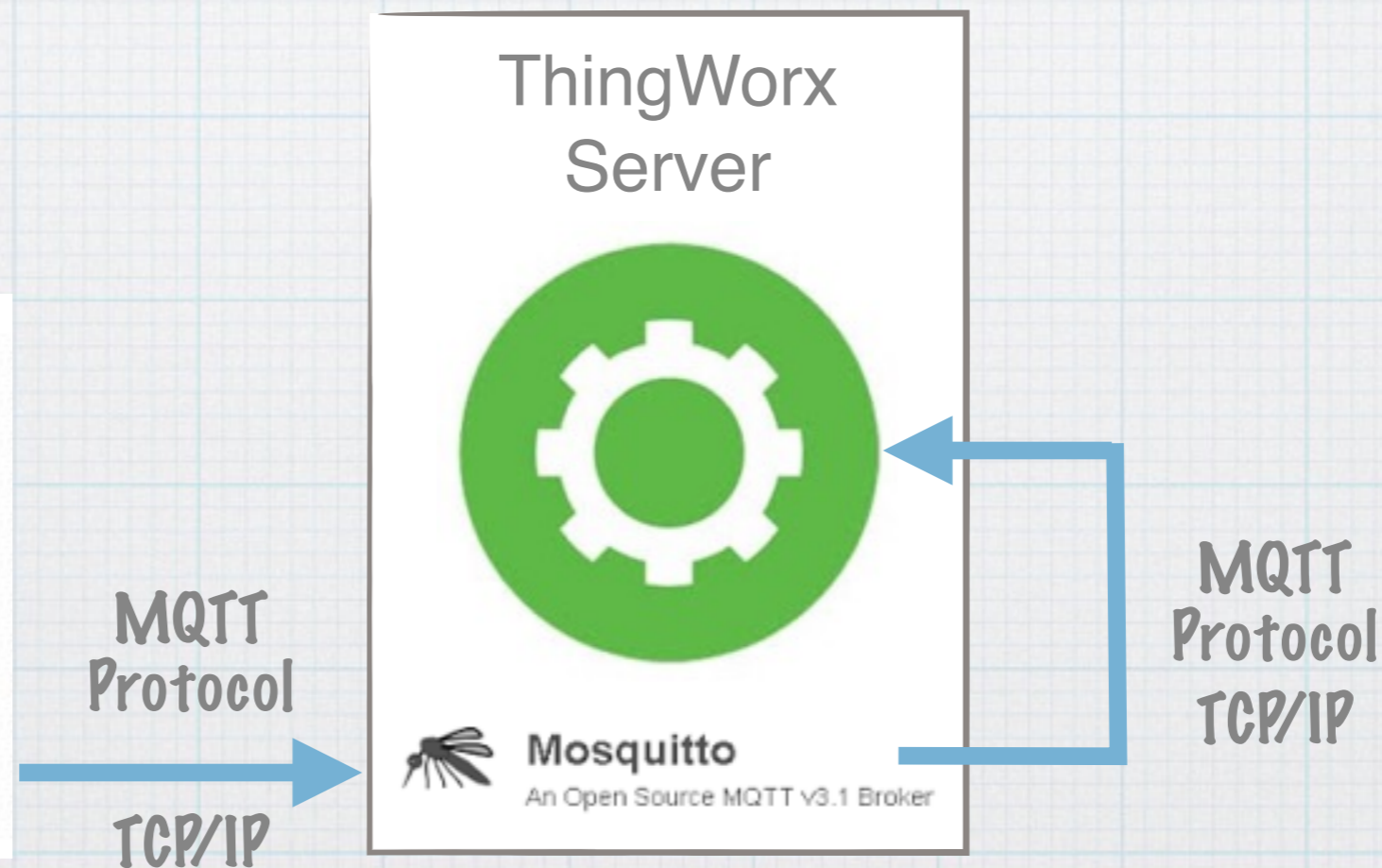
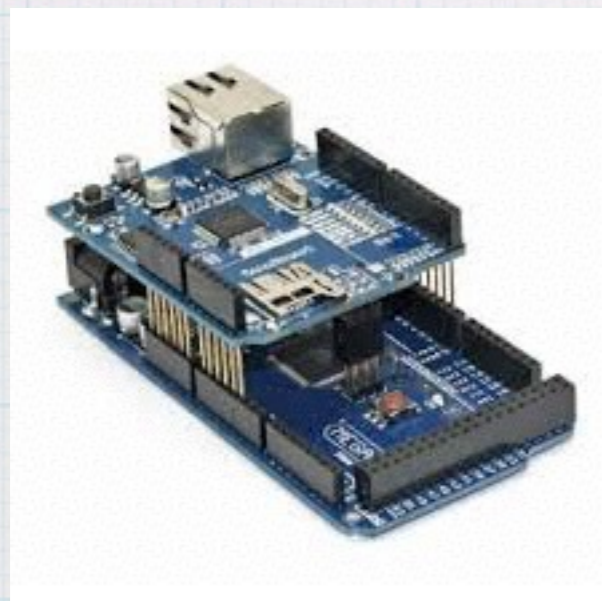
Using MQTT over a Network  
By

Bill Reichardt ([bill.reichardt@thingworx.com](mailto:bill.reichardt@thingworx.com))



# Overview

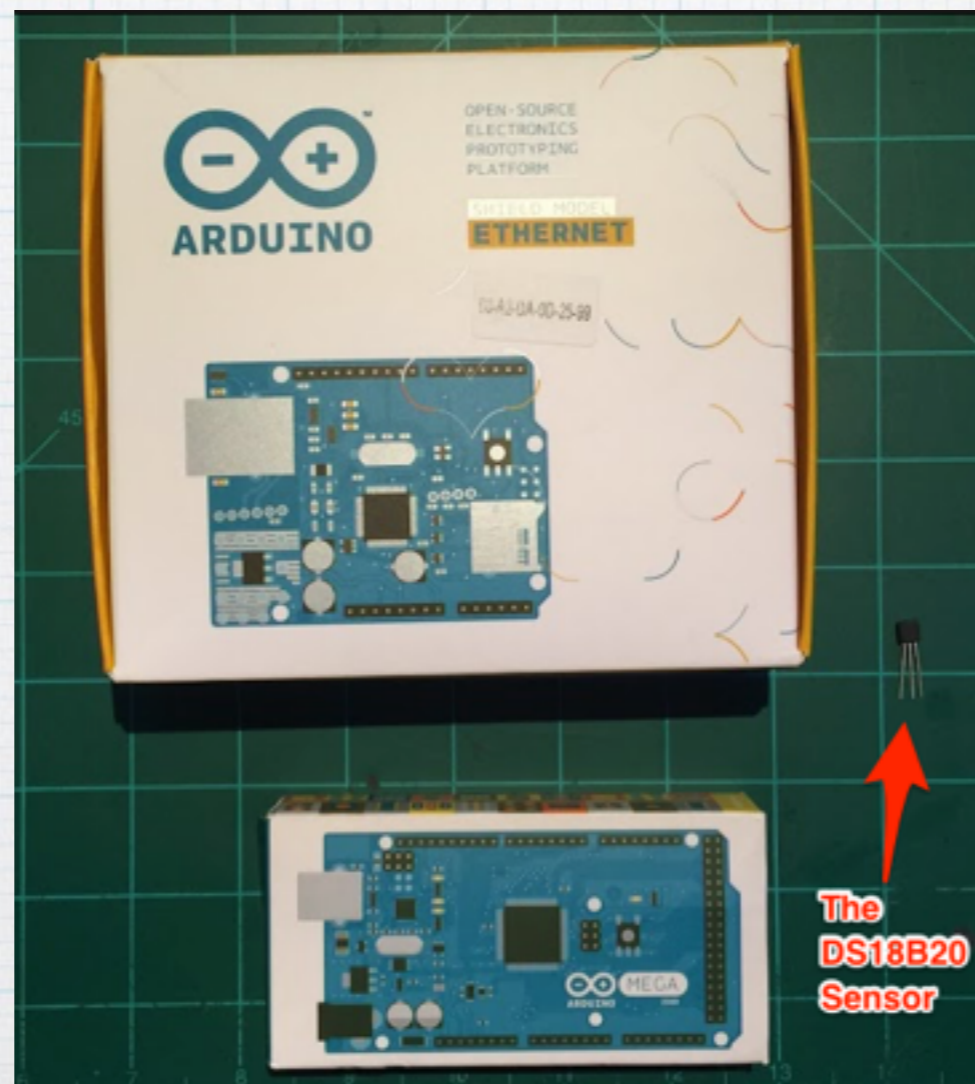
- \* Arduino is a low resource platform – Not suitable for any ThingWorx SDK
- \* MQTT protocol was designed for low resource platforms
- \* ThingWorx provides a free MQTT extension





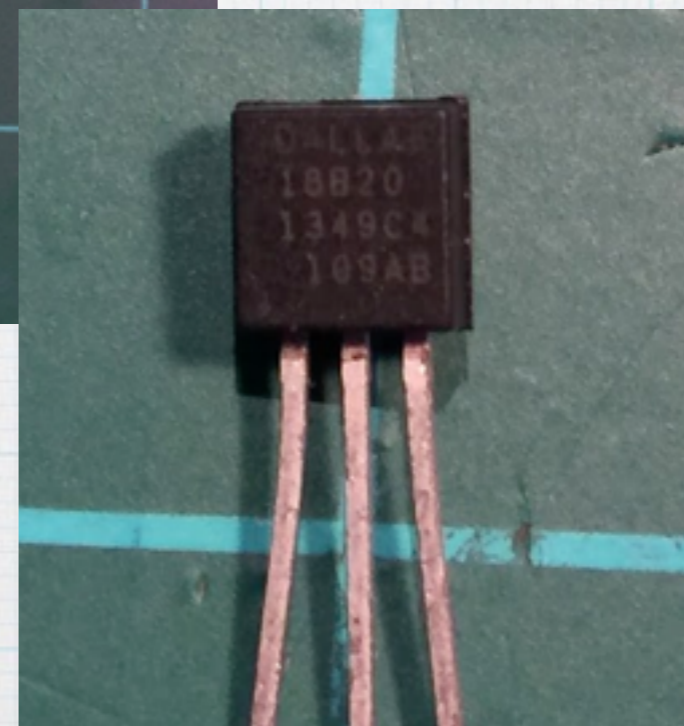
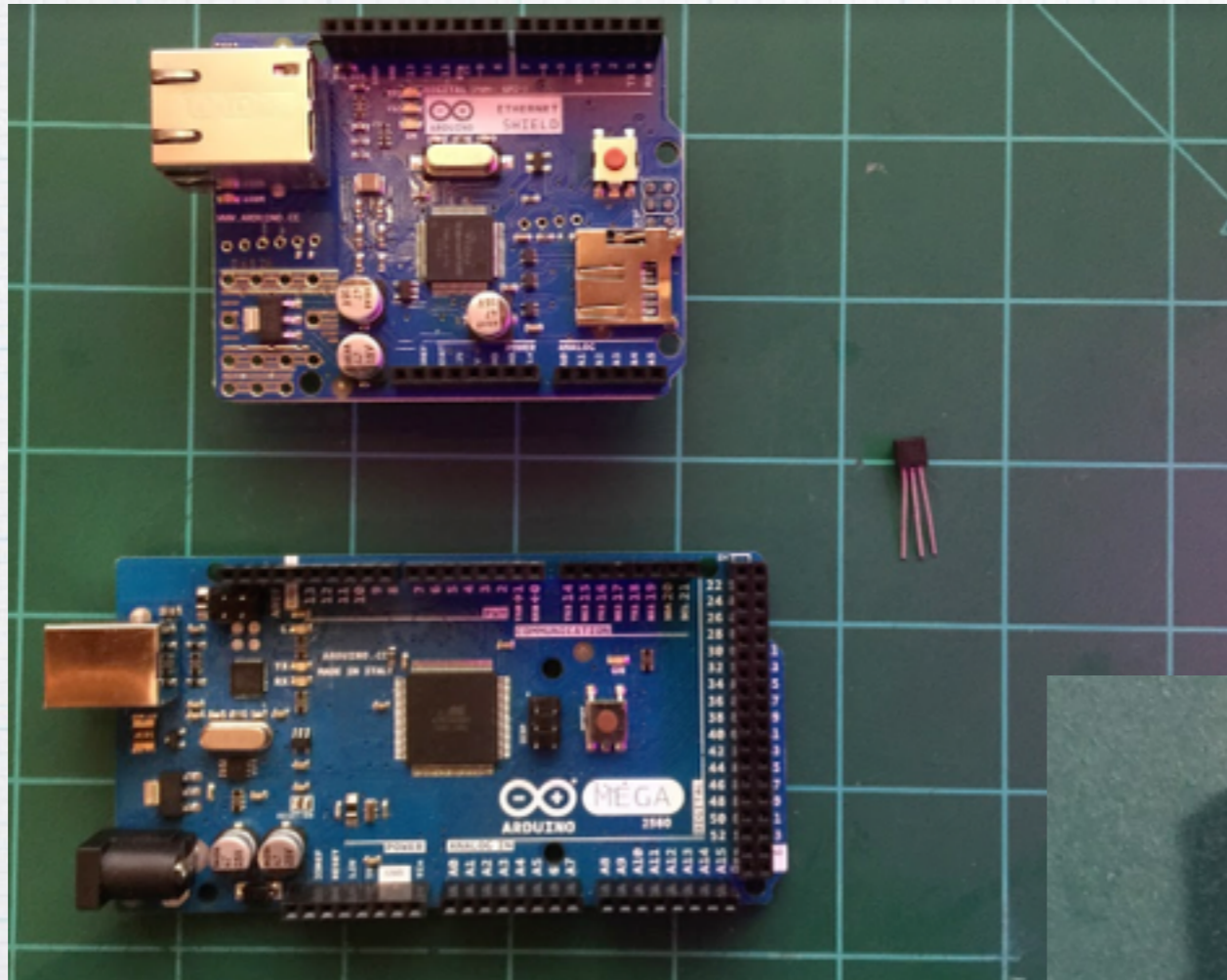
# The Hardware

- \* 1 Arduino, I used the Mega2560
- \* 1 Arduino Ethernet Shield
- \* 1 Dallas Systems DS18B20 "One Wire" Temperature sensor



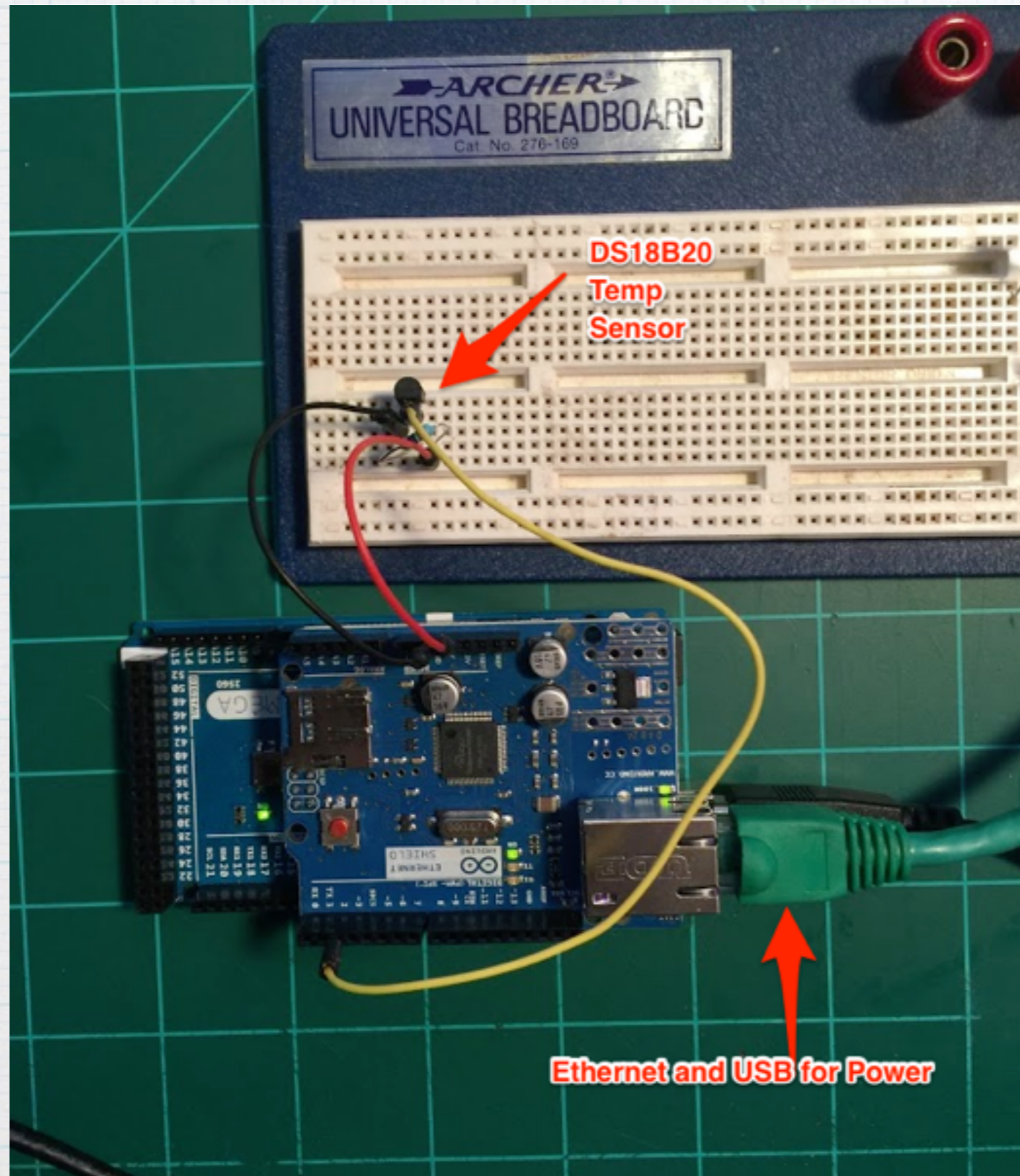


# Hardware Out of the Box

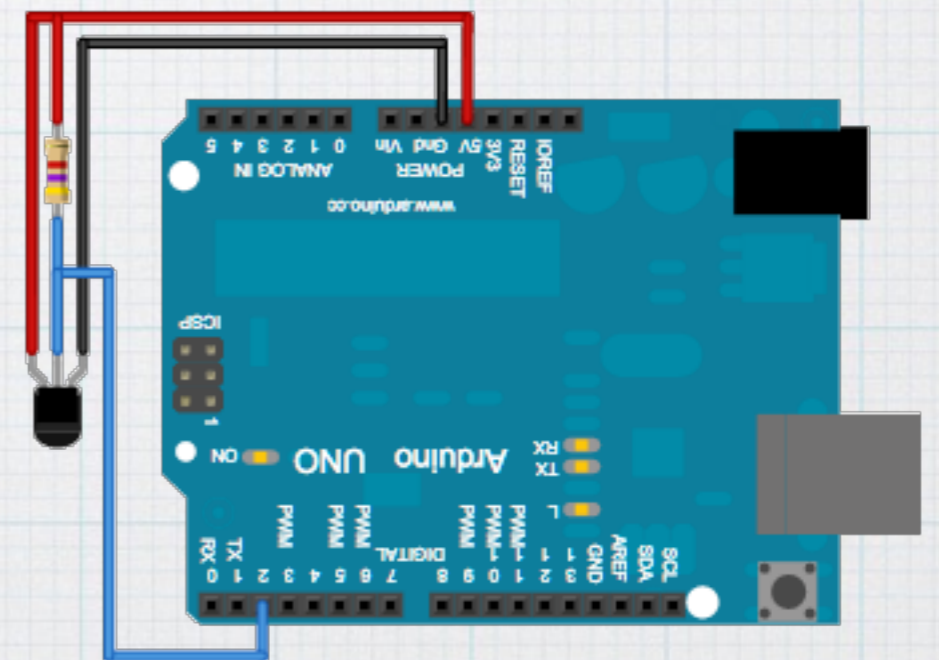




# Assembled



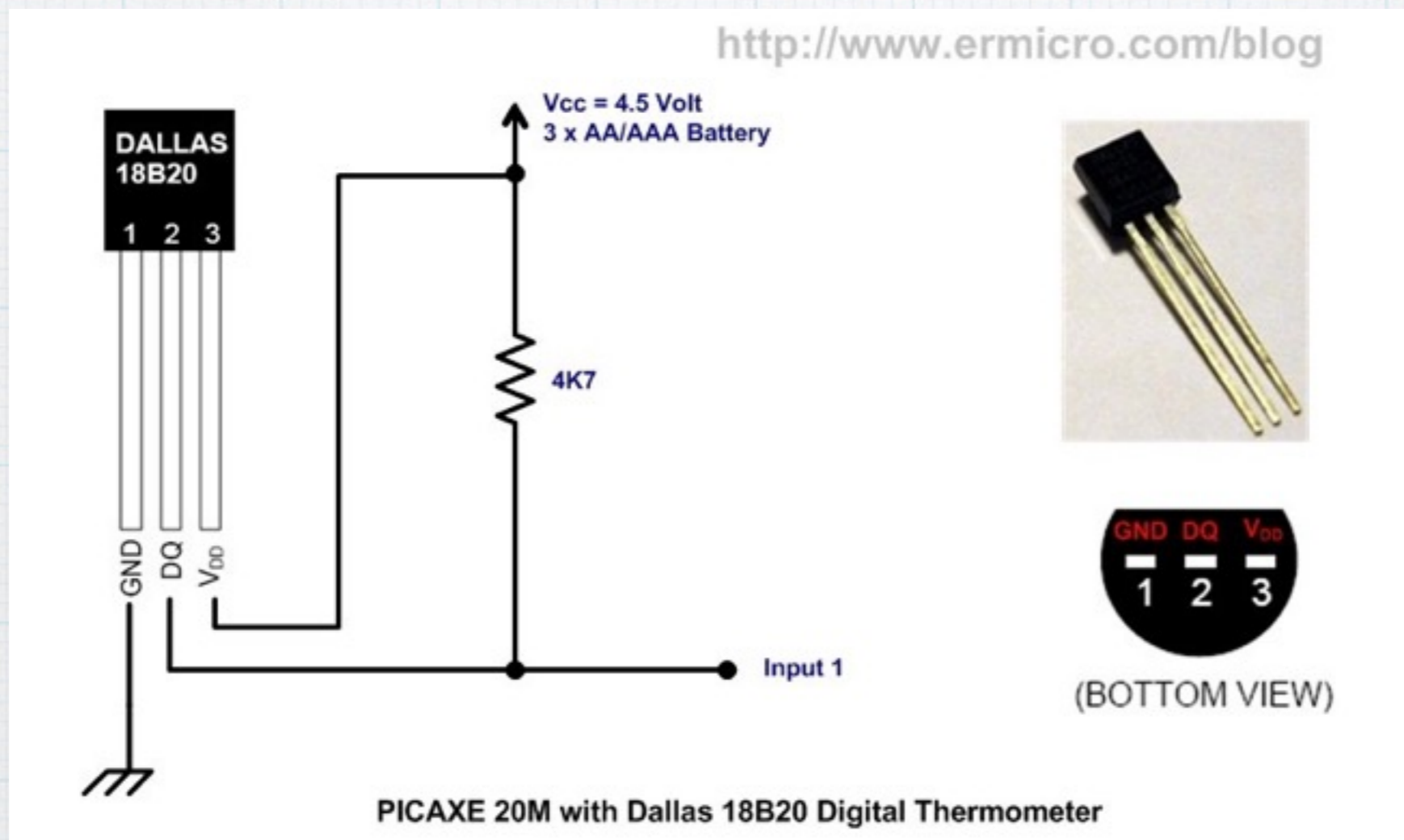
- \* 4.7 K $\Omega$  Resistor is required (see next slide)
- \* The Sensor is powered by the Arduino via USB
- \* Multiple Sensors can share the same pins and resistor





# Warning!

- \* Do not reverse +5V and GND on the 18B20!!!
- \* The sensor will overheat and burn out in seconds
- \* Buy more than one



More detailed description of this hardware can be found at <http://www.hacktronics.com/Tutorials/arduino-1-wire-tutorial.html>



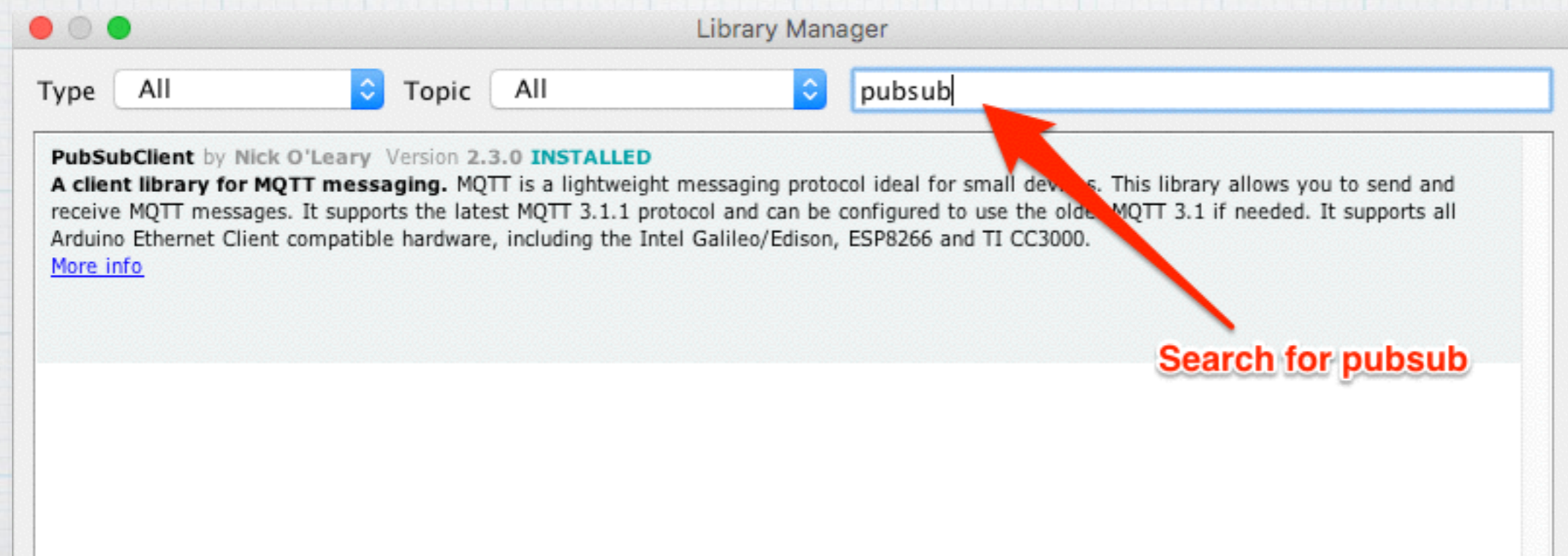
# Create an MQTT Server

- \* An MQTT server stores property updates from devices
- \* ThingWorx can apply these updates to Things on the server
- \* **Mosquitto MQTT server**
  - \* <http://mosquitto.org/>
- \* OSX Install <http://www.xappsoftware.com/wordpress/2014/10/30/install-mosquitto-on-mac-os-x/>
- \* Windows comes with an installer
- \* You can usually install it on the same system that ThingWorx is on.



# Install pubsubclient

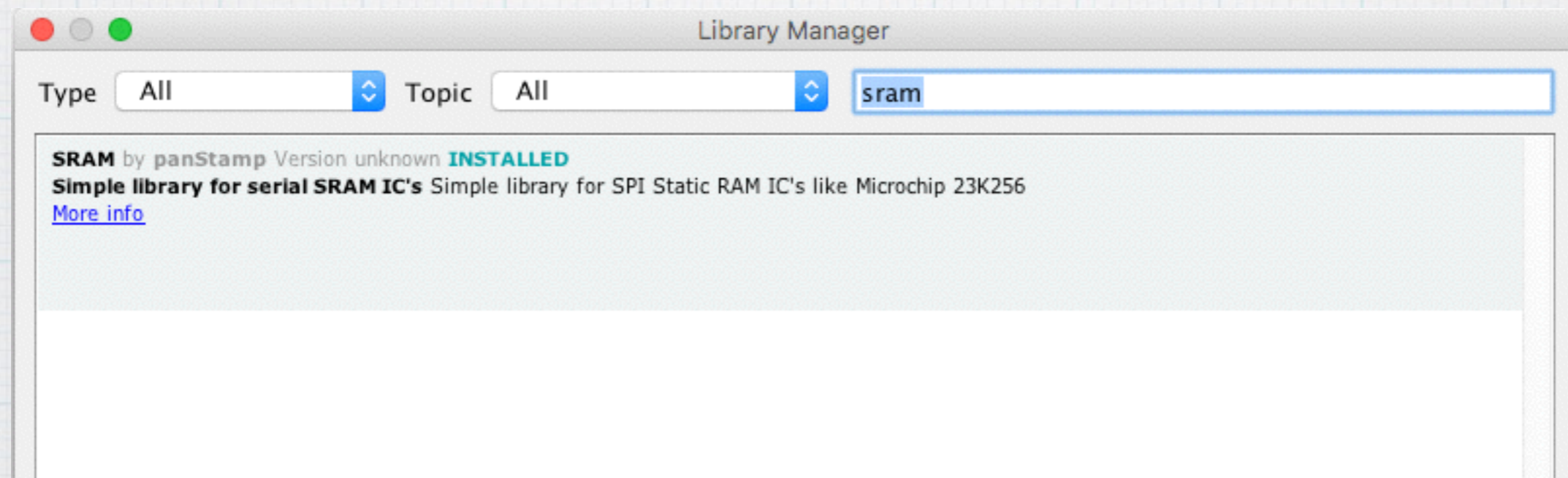
- \* You will need the **pubsubclient** library installed in your Arduino IDE
- \* Use the Sketch>Include Library>Manage Libraries.. Menu to install it





# Install sram

- \* You will need the **sram** library
- \* It is required by pubsub
- \* You may have to restart before you can use your installed libraries





# Test Out Your Connection

- \* In your Arduino IDE menu go to File>Sketchbook>libraries>pubsubclient>mqtt\_stream
- \* This should open the mqtt\_stream.ino example
- \* Make the changes shown here and upload it to your Arduino
- \* Script creates a topic called **outTopic**



```
mqtt_stream | Arduino 1.6.4
mqtt_stream
/*
Example of using a Stream object to store the message payload

Uses SRAM library: https://github.com/ennui2342/arduino-sram
but could use any Stream based class such as SD

- connects to an MQTT server
- publishes "hello world" to the topic "outTopic"
- subscribes to the topic "inTopic"
*/

#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>
#include <SRAM.h>

// Update these with values suitable for your network.
byte mac[] = { 0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
IPAddress ip(172, 16, 0, 100);
IPAddress server(172, 16, 0, 2);

SRAM sram(4, SRAM_1024);

void callback(char* topic, byte* payload, unsigned int length) {
  sram.seek(1);
}
```

**Update the mac using the numbers on your arduino ethernet shield box**

**Pick a static ip address on your network**

**This is the ip address of your matt server**



# Testing Your Server

- \* Use the Mosquitto command line client to subscribe to your topic
- \* Run:
  - \* `mosquitto_sub -h 127.0.0.1 -t outTopic`
- \* `-h` is the server, `127.0.0.1` assumes you are running this command on the same host
- \* If everything is working, you should see "Hello World" in appear in your console when you turn your Arduino on.



# Connecting the Sensor

- \* There should be a `DS18x20_MQTT.ino` sketch accompanying this slide deck
- \* You may have to install the `OneWire` library to compile it
- \* Make the same IP address and Mac changes you made to the `MQTT_Stream.ino` example
- \* Note the topic changes, `outTopic` has been replaced with `/Thingworx/DS18Thing/F & C` to report temp readings

```
if (cfg == 0x00) raw = raw & ~1; // 9 bit resolution, 93.75 ms
else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
//// default is 12 bit resolution, 750 ms conversion time
}
celsius = (float)raw / 16.0;
fahrenheit = celsius * 1.8 + 32.0;
Serial.print(" Temperature = ");
Serial.print(celsius);
Serial.print(" Celsius, ");
Serial.print(fahrenheit);
Serial.println(" Fahrenheit");
char tempString[100];
String(fahrenheit).toCharArray(tempString, sizeof(tempString));
client.publish("/Thingworx/DS18Thing/F", tempString);
String(celsius).toCharArray(tempString, sizeof(tempString));
client.publish("/Thingworx/DS18Thing/C", tempString);
client.loop();
}
```










These topics  
map to  
Thingworx  
properties





# Installing the MQTT Extension

- \* Download and install the MQTT extension from the ThingWorx Marketplace at:
  - \* <http://marketplace.thingworx.com/downloads/mqtt/>
- \* When done you will now have these things...

<input type="checkbox"/>		 MQTTSubscriber	MQTT subscriber interface	 ThingTemplate
<input type="checkbox"/>		 MQTTConnection	MQTT connection interface	 ThingTemplate
<input type="checkbox"/>		 MQTT	MQTT interface	 ThingTemplate



# Creating a Thing for Your Data

- \* Create A new Thing using the MQTT ThingTemplate called DS18Thing

The screenshot displays the ThingWorx user interface for creating a new Thing. The top navigation bar includes the ThingWorx logo, a search bar, and buttons for '+ New Entity' and 'Import'. Below the navigation bar, the main content area is titled 'New Thing' and contains a 'Save' button, 'Cancel Edit', and 'To Do' dropdown. The left sidebar shows a menu of options under 'ENTITY INFORMATION' and 'PERMISSIONS'. The main content area is divided into sections: 'General Information' (selected), 'Properties', 'Services', 'Events', 'Subscriptions', 'Configuration', and 'Home Mashup'. The 'General Information' section contains a form with the following fields:

Field	Value
Name	DS18Thing
Description	
Tags	Search ModelTags or + t
Thing Template	MQTT
Implemented Shapes	Search ThingShapes or +

The right sidebar shows a table with columns for 'Home M', 'Pul', 'Id', 'Last Modifie', and 'Value'.



# Give it Properties

- \* Since we are sending F & C temperature values, Give your thing an F & C number property.

The screenshot shows the 'Properties' section of a Thingiverse interface for a 'DS18Thing'. The left sidebar contains navigation options like 'General Information', 'Properties', 'Services', 'Events', 'Subscriptions', 'Configuration', 'Home Mashup', 'Permissions', 'Visibility', 'Design Time', 'Run Time', 'Change History', and 'Dependencies'. The main content area is titled 'Properties' and includes buttons for 'Add My Property', 'Manage Bindings', 'Edit', 'Delete', and 'Duplicate'. Below this, there are two tables of properties.

**My Properties**

<input type="checkbox"/>	Edit	Name	Type	Alerts	Additional Info	Default Value	Value	<input type="checkbox"/>	DataChange	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		# F		0 Alerts			69.5	<input type="checkbox"/>	Set	VALUE: 0	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		# C		0 Alerts			20.8	<input type="checkbox"/>	Set	VALUE: 0	<input type="checkbox"/>	<input type="checkbox"/>

**MQTT (ThingTemplate) - Properties**

**Connectable**

Name	Type	Alerts	Additional Info	Default Value	Value	<input type="checkbox"/>	DataChange	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
isConnected		0 Alerts		false	true	<input type="checkbox"/>	Set	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
lastConnection		0 Alerts			2015-10-31 15:...	<input type="checkbox"/>	Set	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Generic Properties**

A red arrow points from the text 'Create the F & C Numeric Properties' to the '# F' and '# C' rows in the 'My Properties' table.



# Configure Your Thing

- \* The configuration tab is where you enter your server settings.
- \* Just enter your server ip address.

The screenshot shows the configuration interface for MQTTThing. The left sidebar has the 'Configuration' tab selected. The main area displays the 'Configuration for MQTTThing' settings. A table of 'Connection Settings' is shown with the following data:

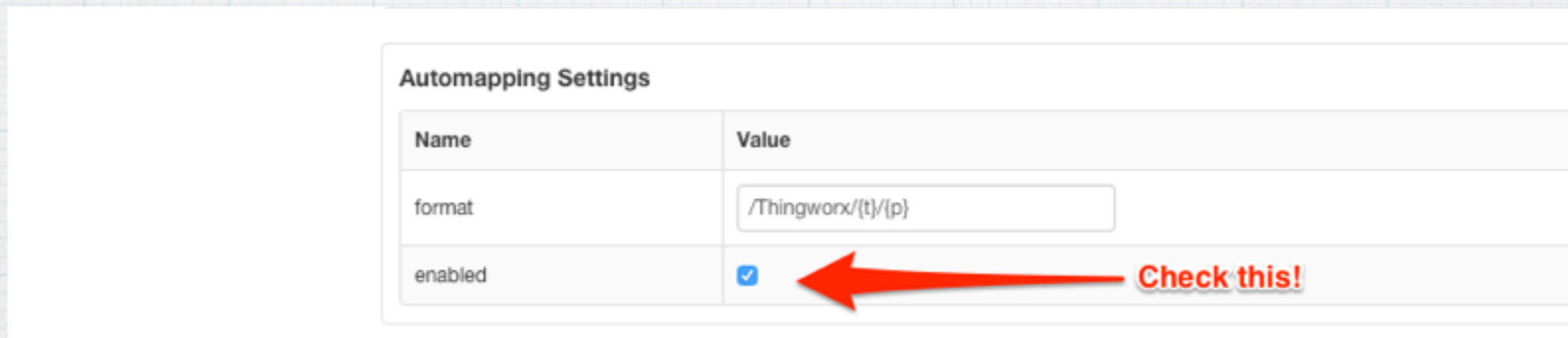
Name	Value
password	<input type="button" value="Change Password"/>
connectTimeout	<input type="text" value="10000"/>
serverName	<input type="text" value="192.168.1.101"/>
retryInterval	<input type="text" value="30000"/>
serverPort	<input type="text" value="1883"/>
userId	<input type="text"/>
timeout	<input type="text" value="5000"/>
useSSL	<input type="checkbox"/>

A red arrow points to the 'serverName' field, which contains the IP address '192.168.1.101'. A red text label 'Your Mosquitto MQTT Server IP Address' is positioned above the arrow.



# Automating Makes it Easy

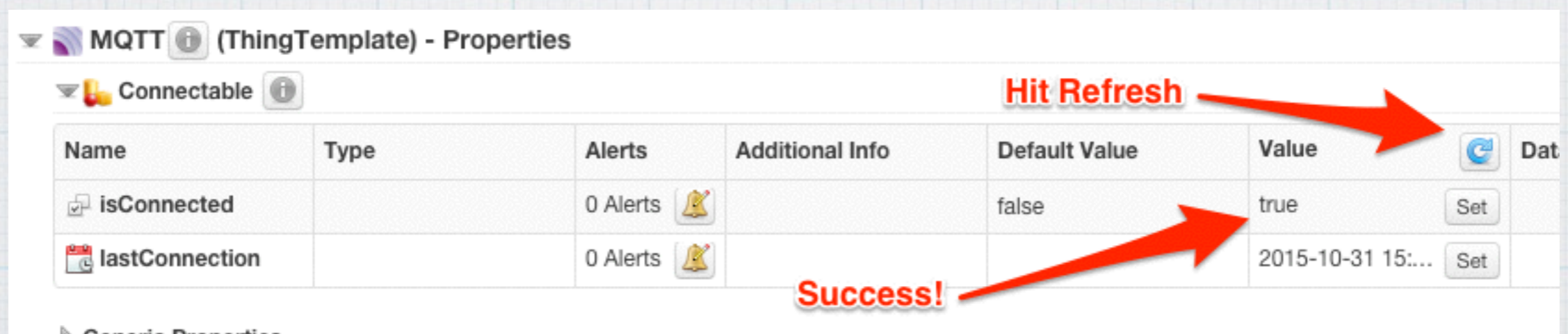
- \* The Default Automapping settings work
- \* The two topics we are using are
  - \* Fahrenheit Temp - /Thingworx/DS18Thing/F
  - \* Celcius Temp - /Thingworx/DS18Thing/C
- \* Generally /Thingworx/<Thingname>/<propname>
- \* Make sure you check enabled!





The screenshot shows the 'Automapping Settings' interface. It contains a table with two columns: 'Name' and 'Value'. The 'format' row has a text input field containing '/Thingworx/{t}/{p}'. The 'enabled' row has a checked checkbox. A red arrow points from the text 'Check this!' to the checked checkbox.

Name	Value
format	<input data-bbox="1275 1195 1783 1257" type="text" value="/Thingworx/{t}/{p}"/>
enabled	<input checked="" data-bbox="1275 1297 1303 1328" type="checkbox"/>

- \* Save this thing. If all is well you will see



The screenshot shows the 'MQTT (ThingTemplate) - Properties' interface. It features a 'Connectable' section with a table of properties. A red arrow points from the text 'Hit Refresh' to a refresh icon in the 'Value' column. Another red arrow points from the text 'Success!' to the 'true' value in the 'Value' column for the 'isConnected' property.

Name	Type	Alerts	Additional Info	Default Value	Value	Buttons	Date
<input checked="" type="checkbox"/> isConnected		0 Alerts 		false	true	<input type="button" value="Set"/>	
<input checked="" type="checkbox"/> lastConnection		0 Alerts 			2015-10-31 15:...	<input type="button" value="Set"/>	



# Watch Your Data Roll In

- \* Once your connection is established, touch the sensor
- \* Hitting the refresh button will allow you to observe the temperature changes

The screenshot shows the ThingWorx interface for a DS18Thing entity. The top navigation bar includes the ThingWorx logo, a search bar, and menu items for New Entity, Import/Export, Monitoring, Help, and Administrator. The main content area is titled 'Properties' and features a table of 'My Properties' and 'MQTT (ThingTemplate) - Properties'.

**My Properties Table:**

Edit	Name	Type	Alerts	Additional Info	Default Value	Value	DataChange
<input type="checkbox"/>	# F		0 Alerts			69.6	Set VALUE: 0
<input type="checkbox"/>	# C		0 Alerts			20.9	Set VALUE: 0

**MQTT (ThingTemplate) - Properties Table:**

Name	Type	Alerts	Additional Info	Default Value	Value	DataChange
isConnected		0 Alerts		false	true	Set
lastConnection		0 Alerts			2015-10-31 15:...	Set

Annotations in the image include a red arrow pointing to the 'Hit Refresh' button and another red arrow pointing to the 'Value' column of the 'My Properties' table, with the text 'These Values will Change' next to it.



# Conclusion

- \* Multiple DS18 Temperature sensors can share the same, single I2C bus on your Arduino so you have created a platform to take multiple temperature Readings
- \* The MQTT server can store readings and deliver them when the ThingWorx server is visible on your network
- \* MQTT protocol is higher performance and easier to create than a REST web service connection
- \* I want to credit a great topic in the ThingWorx community for getting me started on this project.
- \* <https://community.thingworx.com/message/5063#5063>