



MIKE KARLINER

MARCH 30, 2015

How to use a Plain ol' Arduino with ThingStudio



If you would like to try out [ThingStudio \(http://www.thingstud.io\)](http://www.thingstud.io), and don't own the requisite networking hardware, but do have a standard Arduino Uno or other device with just a serial connection, do not despair! We have created a useful program which will get your 'duino running with ThingStudio in no time.

The DesktopMQTT is a simple to install Macintosh application that provides two, independent, functions. First, it gives you your own, local, MQTT broker. Second, it provides a bridge between the serial connection you have to your Arduino (or indeed, anything else with a serial connection), and your MQTT broker, and thus, to ThingStudio. Today we're just going to use both of those functions. Tip: For a quick backgrounder on MQTT you can look [here \(http://www.thingstud.io/docs/about-thingstudio-mqtt\)](http://www.thingstud.io/docs/about-thingstudio-mqtt). For a more permanent MQTT broker which works on Raspberry Pi and other small platforms, checkout mosquitto.org, and use the latest one.

Right, let's get started.

First download the DesktopMQTT installer from here

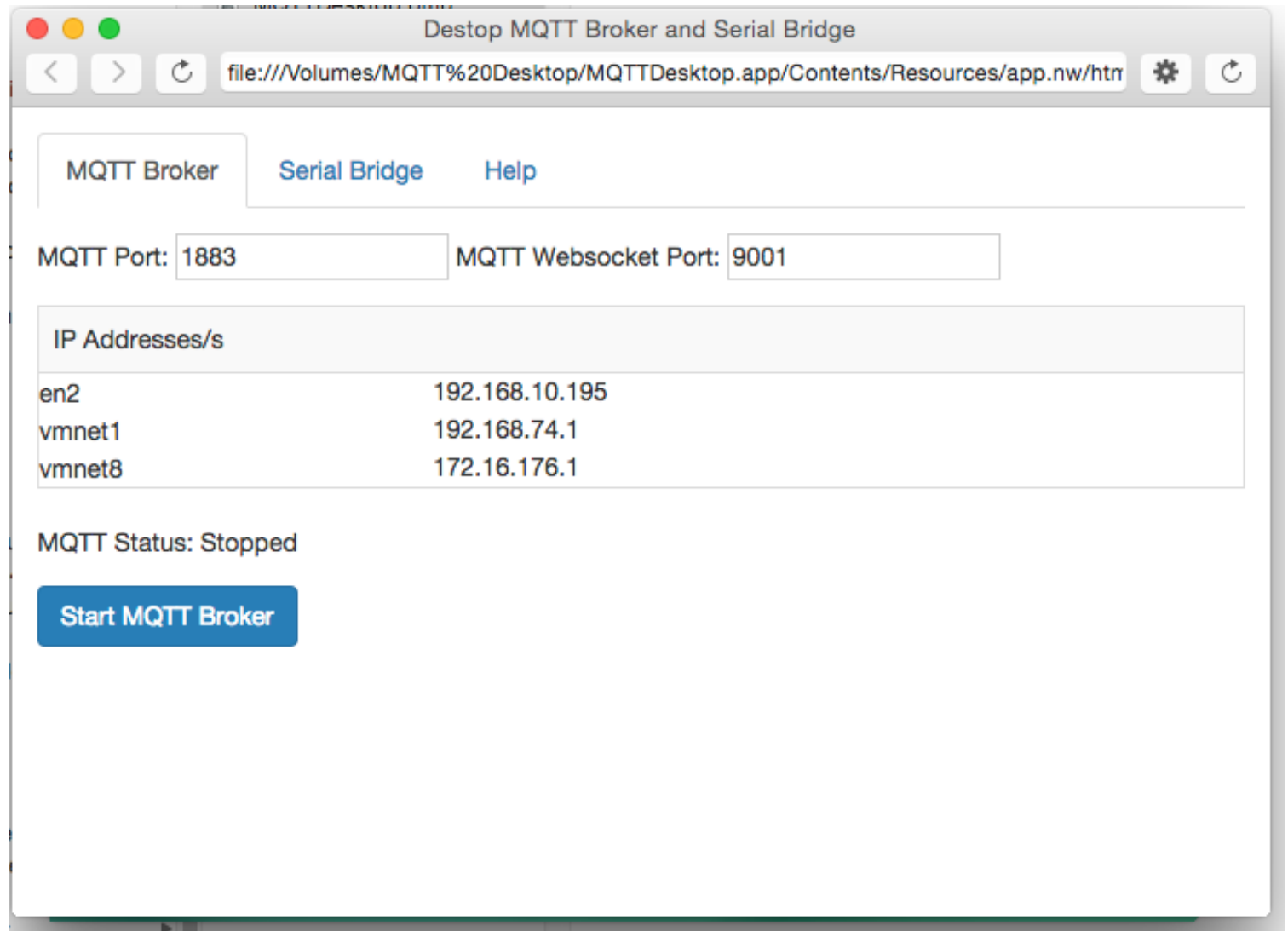
<http://mqtt.thingstud.io/MQTDesktop.dmg>
(<http://mqtt.thingstud.io/MQTDesktop.dmg>)

and drag the application icon to 'Applications'.



Next, start the MQTTDesktop. **Note:** OSX may complain that the application is not from an approved Apple source. You can override this in 'System Settings/Security and Privacy/General' Allow apps downloaded from...

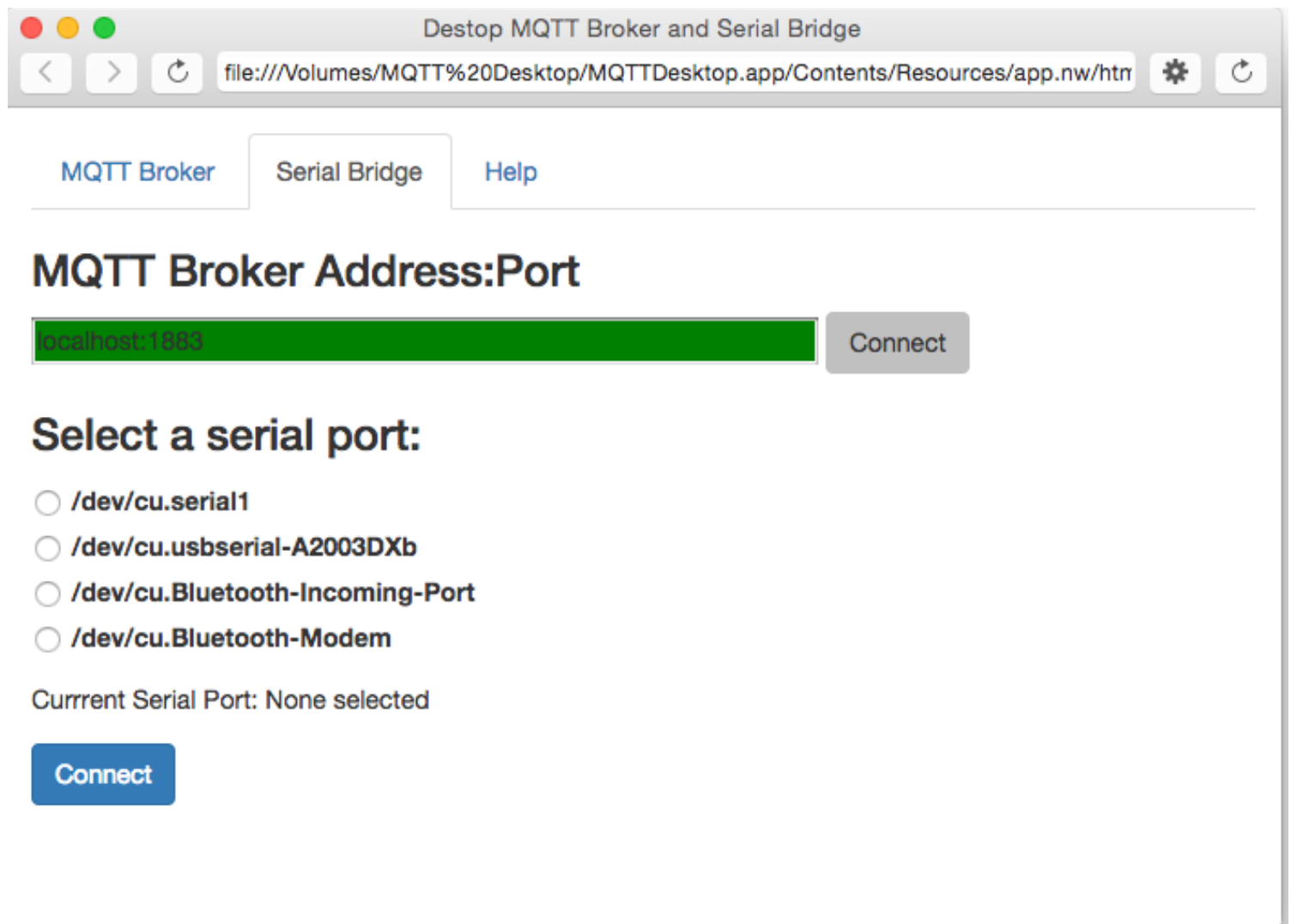
The first tab that shows is the MQTT broker.



(<http://blog.thingstud.io/wp-content/uploads/DesktopMQTT.png>)

Just click on the 'Start MQTT Broker' button to start the broker, and take note of your local IP address, you'll need this when you configure your account at ThingStudio.

Now lets configure your serial bridge.



(http://blog.thingstud.io/wp-content/uploads/Serial_BridgeScreenShot.png)

Check that the MQTT broker text box has a green background. This means that you are connected to the local broker. Next, select the serial port that corresponds to your Arduino.

Hint: This will be the same as the port that you select for the board in the Arduino IDE.

Select the serial port and click connect. The connect button should switch to showing 'Disconnect'.

If it doesn't, remember that only one program can have a serial port open at a time, so if you are using the serial monitor on the Arduino IDE, you won't be able to open the port. This also works the other way around, if you want to upload a new program to the Arduino, you have to disconnect the bridge from the serial port or the upload will fail.

You are now ready to create your first Arduino application that talks to ThingStudio.

You can download all the ThingStudio example programs from github, <https://github.com/mkarliner/ThingStudioExamples.git> (<https://github.com/mkarliner/ThingStudioExamples.git>), but for simplicity, here is a listing of your first sketch:

NOTE: This sketch has been updated to support the new, all JSON ThingStudio message payloads.

```
1  #include <string.h>
2
3  /*
4
5   This example shows how to use the MQTT <--> Serial bridge
6   feature in ThingDesktop.
7
8   You can get this from http://www.thingstud.io/downloads/ThingDesktop.zip
9
10  Usage: flash this to an Arduino, and connect its serial port to the bridge.
11
12  This sketch subscribes to a topic called "ledcmds" and expects messages
13  of 'true' or 'false'. When it receives these it will switch the on-board led
14  on or off and publish its status back to the topic 'ledstatus'
15
16  */
17
18  char inputBuffer[80];
19
20  char *readLine(char *buff) {
21
22      *buff = Serial.read();
```

```

23 while(*buff != '\n'){
24     buff++;
25     while( (*buff = Serial.read()) == -1);
26 }
27 *buff=0;
28 return buff;
29 }
30
31 void setup() {
32     // initialize digital pin 13 as an output.
33     pinMode(13, OUTPUT);
34     Serial.begin(9600);
35     Serial.println("subscribe:ledcmds");
36 }
37
38 int old_val;
39 int diff, val;
40 // the loop function runs over and over again forever
41 void loop() {
42     char *pch;
43
44     if(Serial.available()>0) {
45         readLine(inputBuffer);
46         pch = strtok (inputBuffer,":");
47         pch = strtok(NULL, ":"); // Message
48         //Note the escaped quote marks!
49         // ThingStudio payloads must be JSON!
50         if(strcmp(pch, "\"true\"") == 0) {
51             digitalWrite(13, HIGH);
52             Serial.println("publish:ledstatus:\"on\"");
53         } else {
54             digitalWrite(13, LOW);
55             Serial.println("publish:ledstatus:\"off\"");
56         }
57     }
58 }
59 }

```

Talking to ThingStudio from the Arduino is so simple it doesn't even need a library. There are just two commands that you can send from the Arduino to ThingStudio, publish and subscribe.

Publish: which is of the form *publish:<topic>:<message>*

and

Subscribe: which is of the form *subscribe:<topic>*

ThingStudio will send the Arduino messages like this

<topic>:<message>

So the sketch above does the following:

In setup, it subscribes to the topic `ledcmds` and sets up pin 13, (the standard Arduino led pin) for output.

In the main loop, it tries to read a line from the serial port, and when it has one, it splits the line on the colon character. If the command is 'true' it switches the led on, otherwise it switches it off. It also publishes its status to the 'ledstatus' topic.

Compile the sketch and upload it to the Arduino.

Next, lets check that it works without ThingStudio.

Open your Arduino serial monitor with 'command shift M'.

The Arduino should print:

`subscribe:ledcmds`

Now type '`ledcmds:true`' and enter, and the board led should come on, and the Arduino will print the line '`publish:ledstatus:on`'.

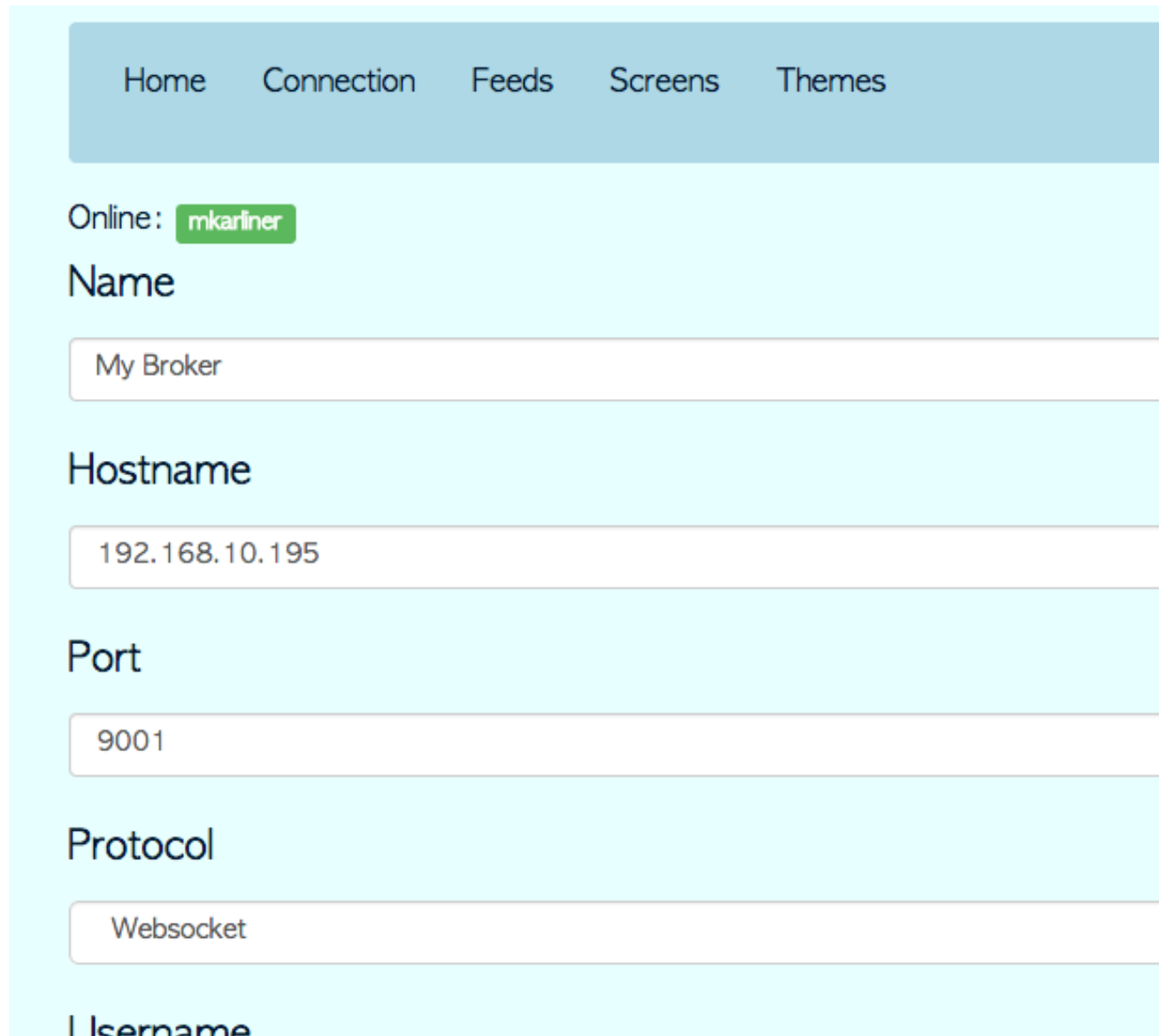
Typing 'ledcmds:false' should turn it off.

Now that we have got the Arduino end working let's write the screen on ThingStudio that drives it.

Sign up to ThingStudio at <http://studio.thingstud.io> (<http://studio.thingstud.io>) if you haven't already done that.

You should start in Studio Mode, this is where you develop your screens. For normal use, you'll use Viewer Mode.

We'll now setup your connection. Click on 'Connection'.



Home Connection Feeds Screens Themes

Online: mkarliner

Name

My Broker

Hostname

192.168.10.195

Port

9001

Protocol

Websocket

Username

(http://blog.thingstud.io/wp-content/uploads/demo_thingstud_io_connection1.png)

In 'Hostname' type the IP address and port of your local broker. Make sure that the protocol is websockets. Type save and make sure that the connection indicator in the top right of your screen goes green to indicate you are connected to your broker. Next we'll set up the feeds that specify the messages to and from your 'duino'.

For this example, we've made things easy and already set up two public feeds for *ledcmds* and *ledstatus* already, so you don't have to create them yourself, but you may want to check them out for when you want to create your own. Click on Feeds, and find and click on the *ledcmds* Edit button. It should look like this.



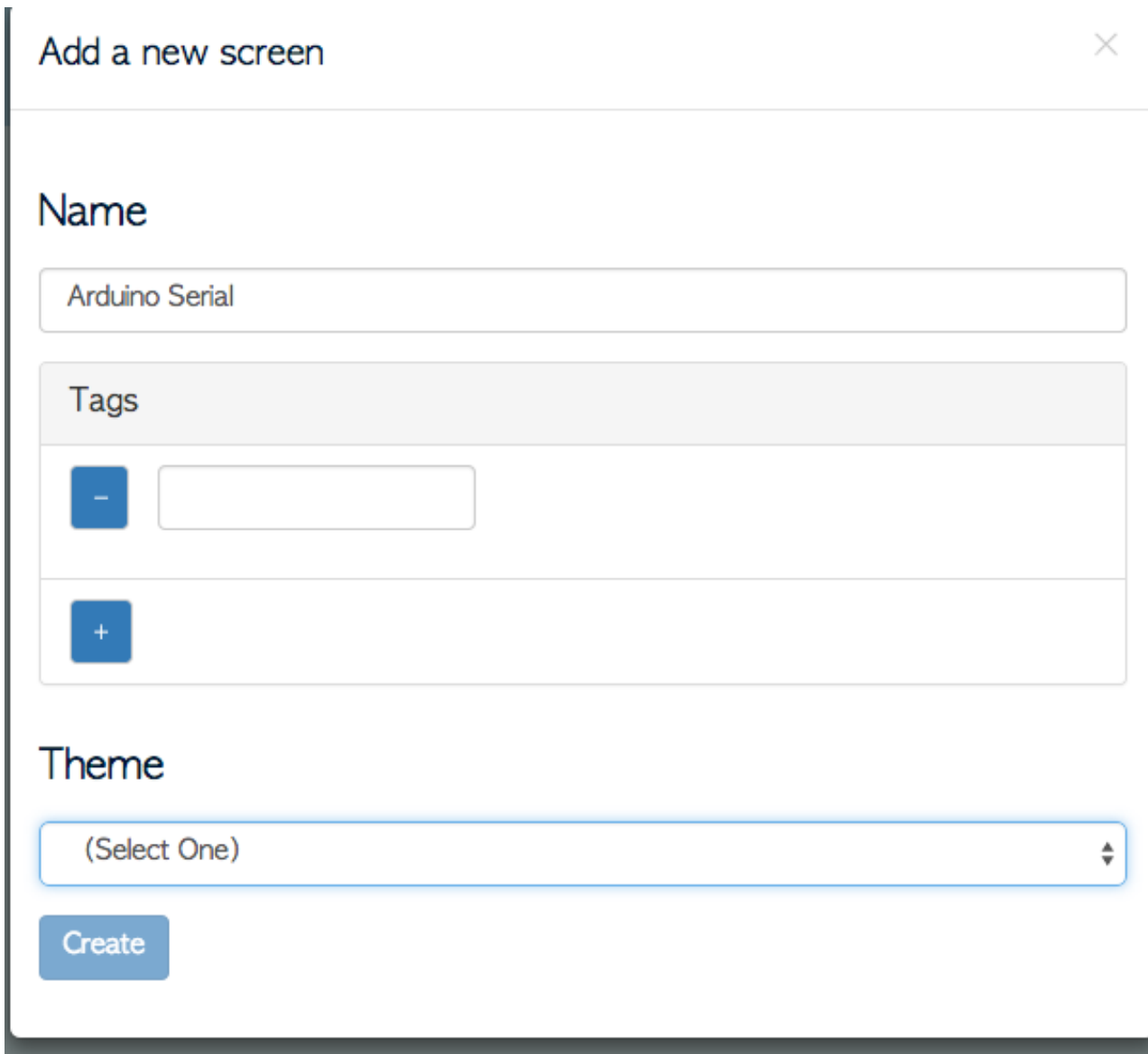
The image shows a web-based form titled "Edit" for configuring a feed. It includes the following elements:

- Name:** A text input field containing the value "ledstatus".
- Subscription:** A text input field containing the value "ledstatus".
- Action:** A dropdown menu with "Update" selected.
- Public:** A checkbox that is checked, with the word "Public" next to it.
- Update:** A blue button at the bottom left of the form.

(http://blog.thingstud.io/wp-content/uploads/demo_thingstud_io_feeds.png)

You don't have to do anything, but this shows you how feeds are defined!

Now to the 'Screens' page. Click on 'Add a new screen'. Just give it a name like 'Arduino Serial' and ignore other fields.



Add a new screen

Name

Arduino Serial

Tags

-

+

Theme

(Select One)

Create

(http://blog.thingstud.io/wp-content/uploads/localhost_3000_screens.png)

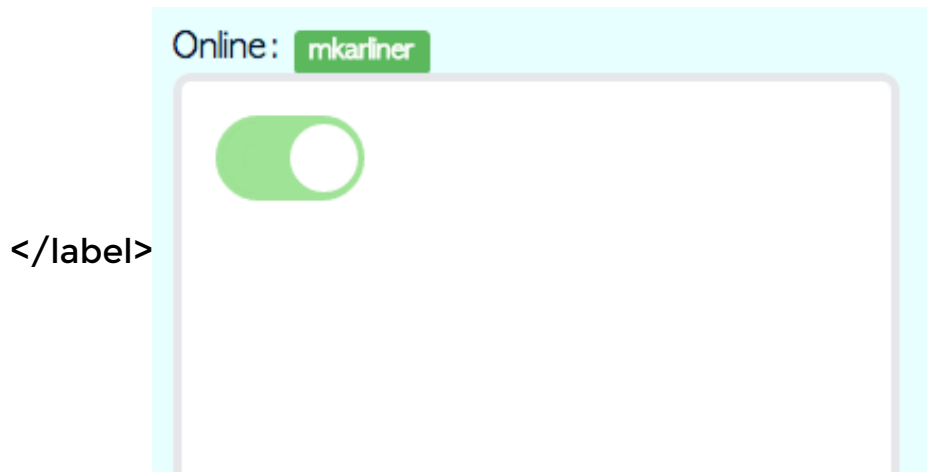
Now, back in the list of screen page, click on your new screen and you'll enter edit mode.

Enter the following code into the edit panel and click update.

```
<label><input type="checkbox" data-feed="ledcmds" class="ios-switch green  
bigswitch" checked />  
<div>
```

```
<div></div>
```

```
</div>
```



[content/uploads/localhost_3000_screens_YJsP5EHxWaLtkve7.png](http://blog.thingstud.io/wp-content/uploads/localhost_3000_screens_YJsP5EHxWaLtkve7.png)

This gives you a nicely styled checkbox. like this:

Now, check the serial bridge is connected, and click on your control. All other things being equal, you should see the led on you Arduino go on. And if you click again, it should go off.

If you have a problem making this work, you can eavesdrop on the MQTT message stream with a tool like MQTT.fx <http://mqttfx.jfx4ee.org/>. Also, the 'Debug' page will show you what MQTT messages ThingStudio is sending and receiving.

OK. Last step, let's also monitor the status of the led that the Arduino is sending back.

Add this to your screen.

```
<p>Led Status: {{message "ledstatus"}}</p>
```

And this will show you the led status as the Arduino switches it on and off.

If you have problems with any of this, you can mail contact@thingstud.io, or use the live chat on the site. Good luck!



About **Mike Karliner** (<http://blog.thingstud.io/author/mkarliner/>)

I make stuff.

TAGS: [arduino](http://blog.thingstud.io/tag/arduino/) (<http://blog.thingstud.io/tag/arduino/>) - [mqtt](http://blog.thingstud.io/tag/mqtt/) (<http://blog.thingstud.io/tag/mqtt/>)

7 Comments



James

April 5, 2015 at 3:16 am (<http://blog.thingstud.io/recipes/how-to-use-a-plain-ol-arduino-with-thingstudio/#comment-2>)

Nice introduction. Had a little problem testing using the Serial Monitor, but the section using ThingStudio worked as expected.

Looking forward to whats to come.



Mike Karliner (<http://www.thingstud.io/>)

April 10, 2015 at 1:48 pm (<http://blog.thingstud.io/recipes/how-to-use-a-plain-ol-arduino-with-thingstudio/#comment-7>)

Glad you like it. We have a lot in the pipeline.



Devesh

April 6, 2015 at 7:42 pm (<http://blog.thingstud.io/recipes/how-to-use-a-plain-ol-arduino-with-thingstudio/#comment-3>)

Is there a windows version of DesktopMQTT ?



Mike Karliner (<http://www.thingstud.io/>)

April 10, 2015 at 1:47 pm (<http://blog.thingstud.io/recipes/how-to-use-a-plain-ol-arduino-with-thingstudio/#comment-6>)

We are trying to get one out very soon.

Mike



Dave Berkeley (<http://www.rotwang.co.uk/projects/projects.html>)

April 7, 2015 at 8:43 am (<http://blog.thingstud.io/recipes/how-to-use-a-plain-ol-arduino-with-thingstudio/#comment-4>)

Interesting stuff. There are a number of problems with the Arduino `readLine()` function.

It can overflow the read buffer. Send it a string > 80 chars and it will crash. You need to pass in the buffer size to the function so it knows when to stop appending chars.

The final `buff++` is wrong. It is already incremented in the loop. This will result in garbage chars being appended to the line.

The `delay(100)` should be unnecessary. At present 2 commands < 100ms apart will get concatenated and one will be ignored. If you intend to dispose of commands, you probably want to act on the most recent command, not the first one in the buffer.

The `"return buff;"` doesn't do anything useful. Why not instead return true if the line is complete, otherwise false? Then you can simply append chars until either the line is full or the line is correctly terminated.

I haven't read further than this first function in the tutorial, or looked at the web code yet.



Mike Karliner (<http://www.thingstud.io/>)

April 10, 2015 at 1:47 pm (<http://blog.thingstud.io/recipes/how-to-use-a-plain-ol-arduino-with-thingstudio/#comment-5>)

Oh, I am embarrassed. I'll get to cleaning this up as soon as I can. Hopefully, you'll understand this was part of the rush to get simple examples out, rather than stupidity in a language that I've been programming for 30 years.



burk

April 16, 2015 at 10:23 pm (<http://blog.thingstud.io/recipes/how-to-use-a-plain-ol-arduino-with-thingstudio/#comment-26>)

Cool demo! I'm looking forward to learning more.

Two problems with this page:

* The arduino code you list above doesn't compile, although SerialBridge.ino from your github works fine. Looks like you inadvertently truncated the first line:

`#include`

* The HTML code you posted doesn't create a checkbox, until you replace the curly quotes with straight ones:

should be:

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

Post Comment

LINKS

[Home \(http://www.thingstud.io\)](http://www.thingstud.io)

[Blog \(http://blog.thingstud.io\)](http://blog.thingstud.io)

[Forum \(http://forum.thingstud.io\)](http://forum.thingstud.io)

[Sign In \(http://www.thingstud.io/sign-in/\)](http://www.thingstud.io/sign-in/)

ABOUT THINGSTUDIO

Create real-time user interfaces for the Internet of Things. [Learn more about ThingStudio \(http://www.thingstud.io\)](http://www.thingstud.io).