Sonoff replacement code

Author David Henry mailto:mgadriver@gmail.com

Download https://github.com/mgaman/Sonoff-modified

Version	Comment	Date
1	Original version	25-Jan-2017

Introduction

1. The Sonoff smart switch is a WiFi controlled switch based on the ESP8266 platform.
2. It can be reprogrammed via the Arduino IDE.
3. The purpose of this code is to make the device controllable by a simple MQTT client that does not require unreasonable access permissions e.g. the Android App MQTT Dash (https://play.google.com/store/apps/details?id=net.routix.mqttdash&hl=en) only requires network access and phone status. The Java client Mqtt.fx (http://mqttfx.jfx4ee.org/) does not require any special access.
4. This code offers capability above simply turning the switch on and off.
5. To make the device portable between various locations I have added the ability to manage an array of WiFi SSID/Password combinations.
6. SSID/Password pairs can be added to or deleted from the device.

Methodology

1. SSID/Password pairs are saved on the local ESP8266 SPIFSS file system.
2. At startup time the device consults the list of local WiFi networks and searches the recorded array for matching SSID's.
3. If there is more than 1 matching SSID the one with the best RSSI is chosen to connect to WiFi.
4. Commands delivered by MQTT performed a variety of functions.
    a. List the current array of SSID's in the device.
    b. List the local WiFi networks available to the device.
    c. Add a new SSID/Password pair.
    d. Delete a SSID/Password pair.
    e. Turn the relay on and off
5. All MQTT commands reply with an indication of the outcome of the order.

Customization

1. The User has to customize the **Sonoff.ino** source file for each separate device.
2. Edit the following #define lines
    a. #define SECURE_MQTT
        i. Leave unchanged if you want a secure SSL connection to the broker. Comment out if you want a non-SSL. Check if your broker supports SSL.
    b. #define DEFAULT_SSID "name"
    c. #define DEFAULT_PWD "9876543210"

i. Define at least one SSID/Password pair that you know will work for the device. This is used to write the default WiFi connection credentials.

d. #define BROKER_ADDRESS "test.mosquitto.org"
   i. The IP address of your broker.
e. #define COMMAND_TOPIC "unique/1/apn/command"
   i. This is the topic for commands flowing from the MQTT client to the device i.e. your MQTT client will **Publish** this topic. Choose a unique name that you are confident will not be used by other devices.
f. #define GENERAL_REPLY_TOPIC "unique/1/apn/replies"
   i. This is the topic for commands flowing from device to the MQTT client i.e. your MQTT client will **Subscribe** to this topic. Choose a unique name that you are confident will not be used by other devices.
g. #define MAX_SSID_LENGTH 50
h. #define MAX_PASSWORD_LENGTH 50
   i. The expected maximum lengths of SSID and Password strings.

Command Language

1. Commands are a single letter followed by optional data.
2. The commands are
   a. I
      i. Initialize the device file system with the DEFAULT_SSID / DEFAULT_PWD pair.
   b. Dssid
      i. ssid is the SSID to be deleted. In practice it is not deleted but just marked as not in use. Ssid is a plain string without surrounding quotes.
   c. ?
      i. List all active SSIDs. Note that passwords are not listed.
   d. L
      i. List all local network SSIDs. Useful for debugging.
   e. A”ssid”,”password”
      i. Add a new SSID/Password pair. The 2 strings must be encased within quotes.
   f. Ron
   g. Roff
      i. Switch the relay on or off.