ColorTouch Remote Control Protocol

Application Programming Interface Guide

v3 rev1

Table of Contents

1. Overview

The Remote Control Protocol enables Skyport WiFi Key equipped ColorTouch thermostats to be controlled via the local network. ColorTouch thermostats available on the local network are discoverable via Simple Service Discovery Protocol (SSDP). The Remote Control Protocol is served by a simple RESTful web service and can be accessed by a wide range of clients written in different programming languages.

ColorTouch thermostats do not allow remote control commands by default. You must go into "Menu" - "Accessories" screen and turn ON "Local API" after connecting to the wireless network.

2. Simple Service Discovery Protocol (SSDP)

The Simple Service Discovery Protocol (SSDP) is a network protocol based on the Internet Protocol Suite for advertisement and discovery of network services and presence information. It accomplishes this without assistance of server-based configuration mechanisms, such as the Dynamic Host Configuration Protocol (DHCP) or the Domain Name System (DNS), and without special static configuration of a network host. SSDP is the basis of the discovery protocol of Universal Plug and Play and is intended for use in residential or small office environments.

SSDP is a text-based protocol based on HTTPU. It uses the User Datagram Protocol (UDP) as the underlying transport protocol. Services are announced by the hosting system with multicast addressing to a specifically designated IP multicast address 239.255.255.250 at UDP port number 1900.

In order to discover the ColorTouch thermostats on your local network, your program can send the following M-SEARCH message using the HTTP protocol to the SSDP multicast address and port.

```
M-SEARCH * HTTP/1.1
Host: 239.255.255.250:1900
Man: ssdp:discover
ST: colortouch:ecp
```

ColorTouch thermostats will respond with the following message when they receive a valid M-SEARCH message.

```
HTTP/1.1 200 OK
Cache-Control: max-age=300
ST: colortouch:ecp
Location: http://192.168.1.100:8080/
USN: ecp:00:23:a7:3a:b2:72:name:Living%20Room
```

Location header is the URL for the ColorTouch Remote Control Service. The thermostat name is contained in the USN header. If you have multiple ColorTouch thermostats on your network, your program will receive multiple packets with different locations and names.

At a regular interval ColorTouch thermostats broadcast their services by sending a NOTIFY message similar to the response message above to the multicast address and port. NOTIFY messages replace the ST header with NT. If your program has not received a new NOTIFY message before the Cache-Control expires, ColorTouch thermostat is no longer available on the network.

3. Remote Control Services

The Remote Control Service is a simple RESTful service available at a HTTP location and port advertised by the SSDP protocol. Communication body structure is standardized on JavaScript Object Notation (JSON).

3.1 Root

This request returns the current API version and the type of ColorTouch thermostat. This command is accessed via an HTTP GET.

Request:
GET / HTTP/1.1

Response:
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 196

```
{
        “api_ver”: 3            // API version
        “type”:”residential”    // Thermostat type: residential or commercial
}
```

3.2 Query

3.1.1 Query / Info

This request returns a list of settings needed to render the home screen of ColorTouch thermostat. This command is accessed via an HTTP GET.

Request:
GET /query/info HTTP/1.1

Response:
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 196

```
{
   “name”:”Thermostat”          // Thermostat name
   "mode": 0,                   // Current thermostat mode
                                //      0: off
                                //      1: heat
                                //      2: cool
                                //      3: auto
   "state": 0,                  // Current thermostat state
                                //      0: idle
                                //      1: heating
                                //      2: cooling
                                //      3: lockout
                                //      4: error
   "fan": 0,                    // Current fan setting
                                //      0: auto
```

```
                                //          1: on
    "fanstate": 0,              // Current fan state
                                //          0: off
                                //          1: on
    "tempunits": 0,             // Current temperature units
                                //          0: fahrenheit
                                //          1: celsius
    "schedule": 0,              // Current schedule state
                                //          0: off
                                //          1: on
    "schedulepart": 0,          // Current schedule part
                                //          0: occupied1 or morning
                                //          1: occupied2 or day
                                //          2: occupied3 or evening
                                //          3: unoccupied or night
                                //        255: inactive
    "away": 0,                  // Current away state (residential only)
                                //          0: home
                                //          1: away
    "holiday": 0,               // Current holiday state (commercial only)
                                //          0: not observing holiday
                                //          1: observing holiday
    "override": 0,              // Current override state (commercial only)
                                //          0: off
                                //          1: on
    "overridetime": 0,          // Time left in override (commercial only)
                                //          0 to 240 minutes
    "forceunocc": 0,            // Current forceunocc state (commercial only)
                                //          0: off
                                //          1: on
    "spacetemp": 73.0,          // Current space temperature
    "heattemp": 70.0,           // Current heat to temperature
    "cooltemp": 75.0,           // Current cool to temperature
    "cooltempmin": 65.0,        // Minimum cool to temperature
    "cooltempmax": 99.0,        // Maximum cool to temperature
    "heattempmin": 35.0,        // Minimum heat to temperature
    "heattempmax": 80.0,        // Maximum heat to temperature
    "setpointdelta": 2.0,       // Minimum temperature difference of heat and cool temperatures
    "hum": 10,                  // Current humidity, if available
    "availablemodes": 0         // Available thermostat modes
                                //          0: all modes
                                //          1: heat only
                                //          2: cool only
                                //          3: heat/cool only
}
```

3.1.2 Query / Sensors

This request returns a list of all the sensors available to the ColorTouch thermostat .This command is accessed via an HTTP GET.

Request:
GET /query/sensors HTTP/1.1

Response:
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 59

```
{
   "sensors": [                    // List of sensors
      {
         "name": "Living Room",    // Sensor name
         "temp": 75.0,             // Sensor temperature
         "hum": 35                 // If available, sensor humidity
      }
   ]
}
```

3.1.3 Query / Runtimes

This request returns the runtime data reports of the ColorTouch thermostat. This command is accessed via an HTTP GET.

Request:
GET /query/runtimes HTTP/1.1

Response:
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 56

```
{
   "runtimes": [                   // List of runtime data
      {
         "ts": "1359403740",       // Timestamp
         "heat1": 5,               // Stage 1 Heat runtime in minutes
         "heat2": 5,               // Stage 2 Heat runtime in minutes
         "cool1": 4                // Stage 1 Cool runtime in minutes
         "cool2": 4                // Stage 2 Cool runtime in minutes
         "aux1": 4                 // Stage 1 Auxiliary runtime in minutes
         "aux2": 4                 // Stage 2 Auxiliary runtime in minutes
         "fc": 4                   // Free Cooling runtime in minutes (residential only)
         "ov": 4                   // Override runtime in minutes (commercial only)
      }
   ]
}
```

3.1.4 Query / Alerts

This request returns a list of alert states for the ColorTouch thermostat. This command is accessed via an HTTP GET.

Request:
GET /query/alerts HTTP/1.1

Response:
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 53

```
{
   "alerts": [                          // List of alerts
      {
         "name": "Service Filter",      // Alert name (Air Filter, UV Lamp, Service Filter)
         "active": false,               // If the alert is active
      }
   ]
}
```

3.2 Control

This request enables remote control of ColorTouch over the network. This command is submitted via an HTTP POST and is URL encoded instead of JSON.

Request:
POST /control HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 45

```
mode=0&                 // Set thermostat mode
                        //      0: off
                        //      1: heat
                        //      2: cool
                        //      3: auto
fan=0&                  // Set fan state
                        //      0: auto
                        //      1: on
heattemp=70&            // Set heat to temperature
cooltemp=75&            // Set cool to temperature
pin=1234                // Passcode, if set on the thermostat
```

Response:
HTTP/1.1 200 OK
Content-Length: 17
{“success”: true}

3.3 Settings

This request enables changing settings of ColorTouch over the network. This command is submitted via an HTTP POST and is URL encoded instead of JSON.

Request:
POST /settings HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
tempunits=0& // Set thermostat temperature units
// 0: Fahrenheit
// 1: Celsius
away=0& // Set away state
// 0: home
// 1: away
schedule=0& // Set schedule state
// 0: off
// 1: on
pin=1234 // Passcode, if set on the thermostat

Response:
HTTP/1.1 200 OK
Content-Length: 17
{“success”: true}

3.4 Error Handling

All the requests in this section returns the described JSON objects if the action was performed without an error. If the request was unsuccessful, the response object is described below.

Error Reponse:
HTTP/1.1 200 OK
Content-Length: 39
```
{
  "error": true,
  “reason”:”error reason”
}
```