

# ESP 8266

Setup

# The chip

- ESP 8266 is the most common wifi chip there is
- There are multiple boards that use ESP chips and are as follows



ESP-01



ESP-02



ESP-03



ESP-04



ESP-05



ESP-06



ESP-07



ESP-08



ESP-09



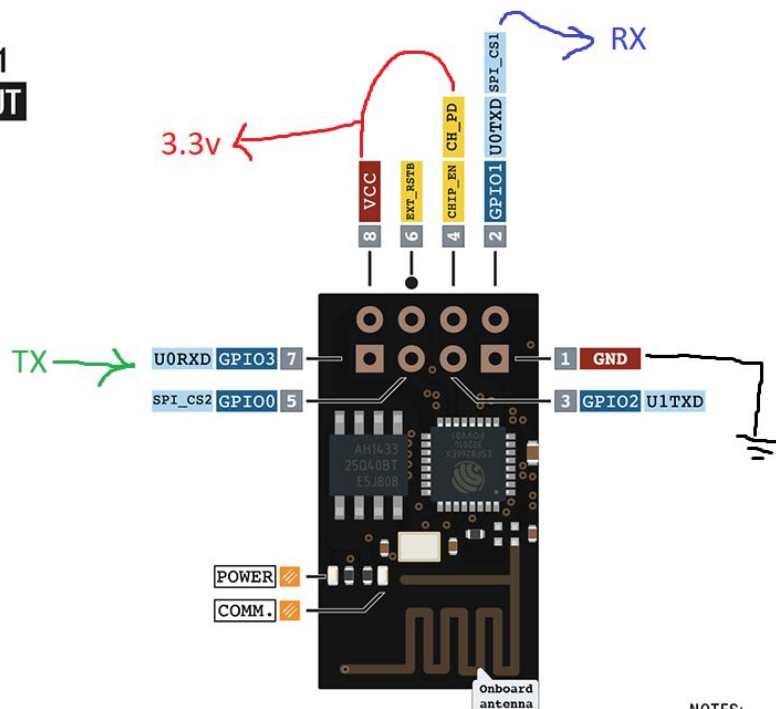
ESP-10



ESP-11

# Pinout and Connections

## ESP-01 PINOUT

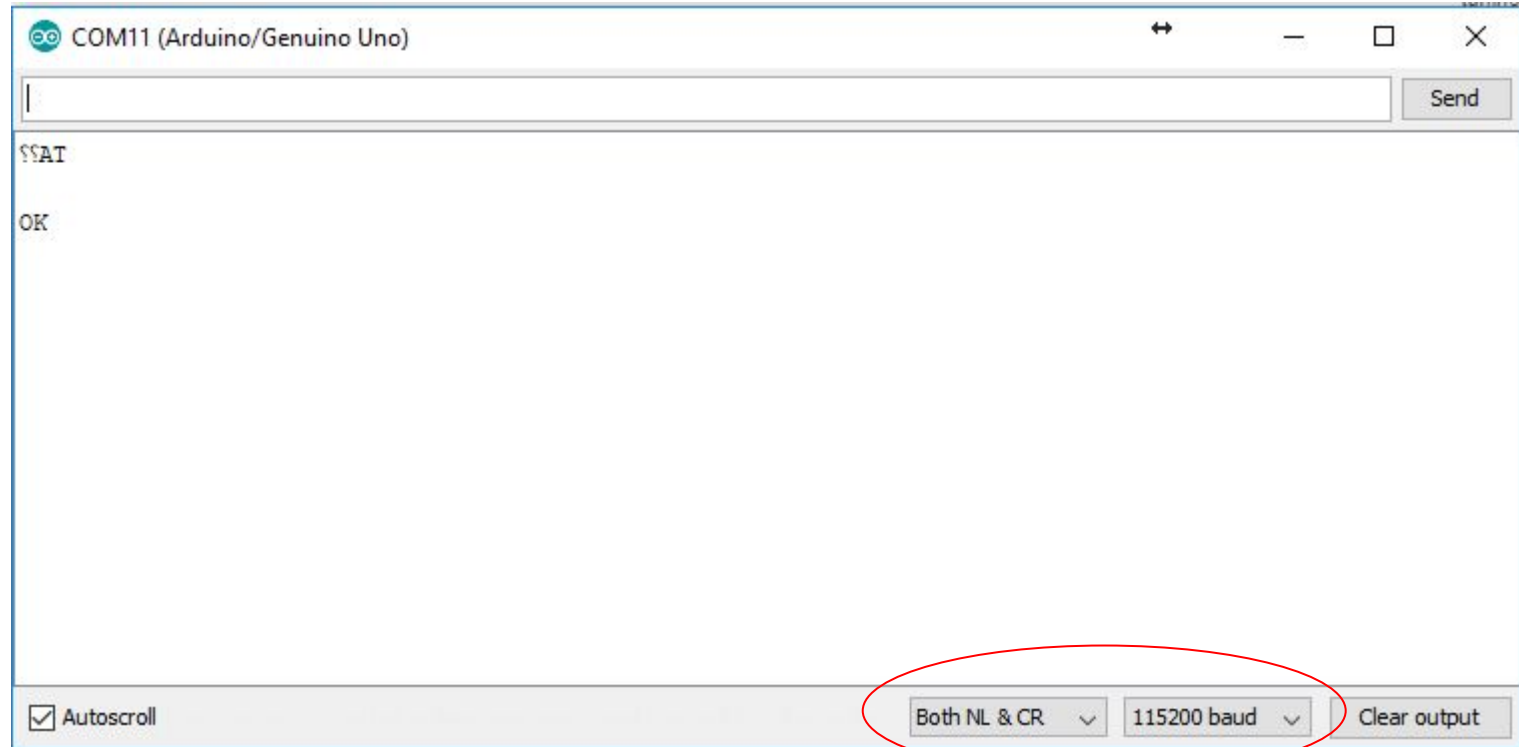


POWER	SP. FUNCTION(S)
I/O	COMM. INTERFACE
ADC	PIN NUMBER
CONTROL	PWM
N/C	

### NOTES:

- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT\_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.

# AT commands in Serial Monitor



# Basic Commands

Commands	Description
AT	Tests AT startup.
AT+RST	Restarts the module.
AT+GMR	Checks version information.
AT+GSLP	Enters Deep-sleep mode.
ATE	Configures echoing of AT commands.
AT+RESTORE	Restores the factory default settings of the module.
AT+UART	UART configuration. <span>@deprecated</span>
AT+UART_CUR	The current UART configuration.
AT+UART_DEF	The default UART configuration, saved in flash.
AT+SLEEP	Configures the sleep modes.
AT+WAKEUPGPIO	Configures a GPIO to wake ESP8266 up from Light-sleep mode.

AT+RFPOWER	Sets the maximum value of the RF TX Power.
AT+RFVDD	Sets the RF TX Power according to VDD33.
AT+RFAUTOTRACE	Sets RF frequency offset trace.
AT+SYSRAM	Checks the available RAM size.
AT+SYSADC	Checks the ADC value.
AT+SYSIOSETCFG	Sets configuration of IO pins.
AT+SYSIOGETCFG	Gets configuration of IO pins.
AT+SYSGPIODIR	Configures the direction of GPIO.
AT+SYSGPIOWRITE	Configures the GPIO output level
AT+SYSGPIOREAD	Checks the GPIO input level.

# WiFi Commands

Commands	Description
AT+CWMODE	Sets the Wi-Fi mode (Station/AP/Station+AP). <b>[@deprecated]</b>
AT+CWMODE_CUR	Sets the Wi-Fi mode (Station/AP/Station+AP); configuration not saved in the flash.
AT+CWMODE_DEF	Sets the default Wi-Fi mode (Station/AP/Station+AP); configuration saved in the flash.
AT+CWLJAP	Connect to an AP. <b>[@deprecated]</b>
AT+CWLJAP_CUR	Connects to an AP; configuration not saved in the flash.
AT+CWLJAP_DEF	Connects to an AP; configuration saved in the flash.
AT+CWLAPOPT	Sets the configuration of command AT+CWLAP.
AT+CWLAP	Lists available APs.
AT+CWQAP	Disconnects from an AP.
AT+CWSAP	Sets the configuration of the ESP8266 SoftAP. <b>[@deprecated]</b>
AT+CWSAP_CUR	Sets the current configuration of the ESP8266 SoftAP; configuration not saved in the flash.

# WiFi Commands

AT+CWSAP_DEF	Sets the configuration of the ESP8266 SoftAP; configuration saved in the flash.
AT+CWLIF	Gets the Station IP to which the ESP8266 SoftAP is connected.
AT+CWDHCP	Enables/Disables DHCP. <span style="color: red;">[@deprecated]</span>
AT+CWDHCP_CUR	Enables/Disables DHCP; configuration not saved in the flash.
AT+CWDHCP_DEF	Enable/Disable DHCP; configuration saved in the flash.
AT+CWDHCP_CUR	Sets the IP range of the DHCP server; configuration not saved in the flash.
AT+CWDHCP_DEF	Sets the IP range of the DHCP server; configuration saved in the flash.
AT+CWAUTOCONN	Connects to an AP automatically on power-up.
AT+CIPSTAMAC	Sets the MAC address of the ESP8266 Station. <span style="color: red;">[@deprecated]</span>
AT+CIPSTAMAC_CUR	Sets the MAC address of the ESP8266 Station; configuration not saved in the flash.
AT+CIPSTAMAC_DEF	Sets the MAC address of ESP8266 station; configuration saved in the flash.

# WiFi Commands

AT+CIPAPMAC	Sets the MAC address of the ESP8266 SoftAP. <b>[@deprecated]</b>
AT+CIPAPMAC_CUR	Sets the MAC address of the ESP8266 SoftAP; configuration not saved in the flash.
AT+CIPAPMAC_DEF	Sets the MAC address of the ESP8266 SoftAP; configuration saved in the flash.
AT+CIPSTA	Sets the IP address of the ESP8266 Station. <b>[@deprecated]</b>
AT+CIPSTA_CUR	Sets the IP address of the ESP8266 Station; configuration not saved in the flash.
AT+CIPSTA_DEF	Sets the IP address of the ESP8266 Station; configuration saved in the flash.
AT+CIPAP	Sets the IP address of ESP8266 SoftAP. <b>[@deprecated]</b>
AT+CIPAP_CUR	Sets the IP address of ESP8266 SoftAP; configuration not saved in the flash.
AT+CIPAP_DEF	Sets the IP address of ESP8266 SoftAP; configuration saved in the flash.
AT+CWSTARTSMART	Starts SmartConfig.
AT+CWSTOPSMART	Stops SmartConfig.
AT+CWSTARTDISCOVER	Enables the mode that ESP8266 can be found by WeChat.
AT+CWSTOPDISCOVER	Disables the mode that ESP8266 can be found by WeChat.
AT+WPS	Sets the WPS function.
AT+MDNS	Sets the MDNS function.
AT+CWHOSTNAME	Sets the host name of the ESP8266 Station.



# Modes - make it station

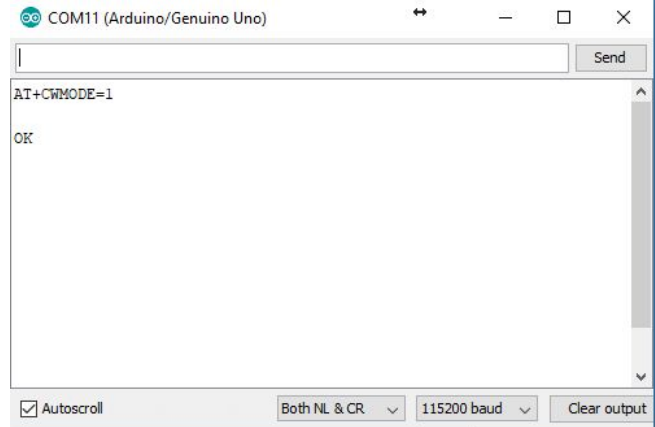
The chip can be made to work in three modes

1. Station
2. Soft AP
3. Station + soft AP

## 4.2.1. AT+CWMODE—Sets the Wi-Fi Mode (Station/SoftAP/Station+SoftAP)

[@deprecated] This command is deprecated. Please use AT+CWMODE\_CUR or AT+CWMODE\_DEF instead.

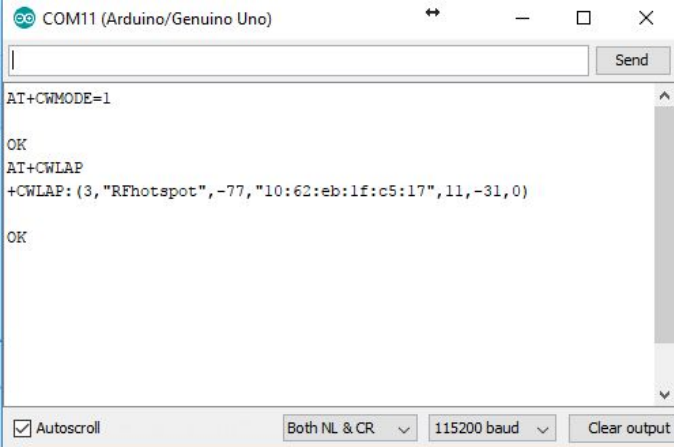
Commands	Test Command: AT+CWMODE=?	Query Command: AT+CWMODE?  Function: to query the current Wi-Fi mode of ESP8266.	Set Command: AT+CWMODE=<mode>  Function: to set the current Wi-Fi mode of ESP8266.
Response	+CWMODE : <mode> OK	+CWMODE : <mode> OK	OK
Parameters	<mode>: ‣ 1: Station mode ‣ 2: SoftAP mode ‣ 3: SoftAP+Station mode		
Note	The configuration changes will be saved in the system parameter area in the flash.		
Example	AT+CWMODE=3		



# Search for WiFi Networks

## 4.2.8. AT+CWLAP—Lists Available APs

Commands	<p>Set Command:</p> <p>AT+CWLAP=&lt;ssid&gt;[,&lt;mac&gt;,&lt;ch&gt;]</p> <p>Function: to query the APs with specific SSID and MAC on a specific channel.</p>	<p>Execute Command:</p> <p>AT+CWLAP</p> <p>Function: to list all available APs.</p>
Response	<p>+CWLAP:&lt;ecn&gt;,&lt;ssid&gt;,&lt;rssi&gt;,&lt;mac&gt;,&lt;ch&gt;,&lt;freq offset&gt;,&lt;freq calibration&gt;</p> <p>OK</p> <p>or</p> <p>ERROR</p>	<p>+CWLAP:&lt;ecn&gt;,&lt;ssid&gt;,&lt;rssi&gt;,&lt;mac&gt;,&lt;ch&gt;,&lt;freq offset&gt;,&lt;freq calibration&gt;</p> <p>OK</p>
Parameters	<ul style="list-style-type: none"><li>• &lt;ecn&gt;: encryption method.<ul style="list-style-type: none"><li>▶ 0: OPEN</li><li>▶ 1: WEP</li><li>▶ 2: WPA_PSK</li><li>▶ 3: WPA2_PSK</li><li>▶ 4: WPA_WPA2_PSK</li><li>▶ 5: WPA2_Enterprise (AT can NOT connect to WPA2_Enterprise AP for now.)</li></ul></li><li>• &lt;ssid&gt;: string parameter, SSID of the AP.</li><li>• &lt;rssi&gt;: signal strength.</li><li>• &lt;mac&gt;: string parameter, MAC address of the AP.</li><li>• &lt;freq offset&gt;: frequency offset of AP; unit: KHz. The value of ppm is &lt;freq offset&gt;/2.4.</li><li>• &lt;freq calibration&gt;: calibration for frequency offset.</li></ul>	
Examples	<p>AT+CWLAP="Wi-Fi","ca:d7:19:d8:a6:44",6</p> <p>or search for APs with a designated SSID:</p> <p>AT+CWLAP="Wi-Fi"</p>	

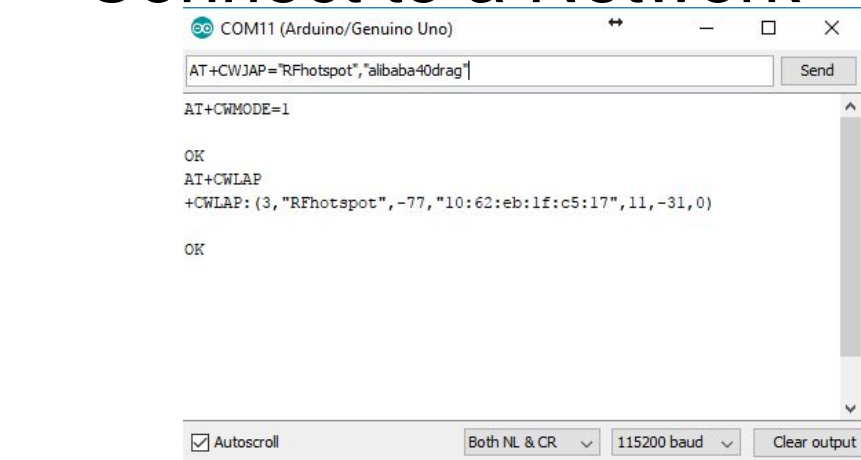


```
COM11 (Arduino/Genuino Uno)
Send
AT+CWMODE=1
OK
AT+CWLAP
+CWLAP: (3, "RHotspot", -77, "10:62:eb:1f:c5:17", 11, -31, 0)
OK
Autoscroll Both NL & CR 115200 baud Clear output
```

# Connect to a Network

## 4.2.4. AT+CWJAP—Connects to an AP

[@deprecated] This command is deprecated. Please use AT+CWJAP\_CUR or AT+CWJAP\_DEF instead.



```
COM11 (Arduino/Genuino Uno)

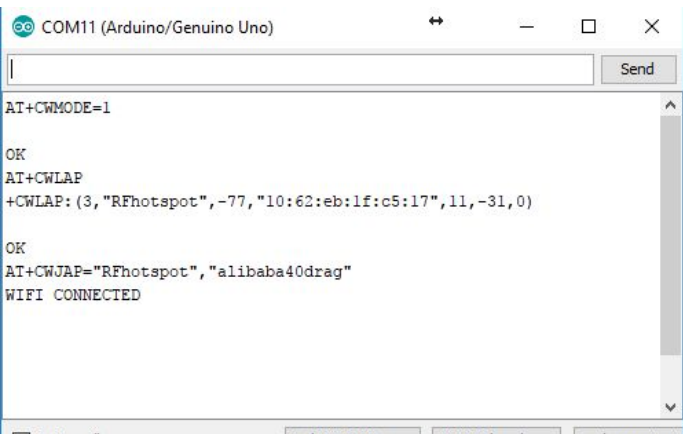
AT+CWJAP="RHotspot","alibaba40drag"

Send

AT+CWJAP=1

OK
AT+CWLAP
+CWLAP: (3, "RHotspot", -77, "10:62:eb:1f:c5:17", 11, -31, 0)

OK
```



```
COM11 (Arduino/Genuino Uno)

AT+CWJAP="RHotspot","alibaba40drag"

Send

AT+CWJAP=1

OK
AT+CWLAP
+CWLAP: (3, "RHotspot", -77, "10:62:eb:1f:c5:17", 11, -31, 0)

OK
AT+CWJAP="RHotspot","alibaba40drag"
WIFI CONNECTED
```

Commands	Query Command: AT+CWJAP?  Function: to query the AP to which the ESP8266 Station is already connected.	Set Command: AT+CWJAP=<ssid>,<pwd>[,<bssid>]  Function: to set the AP to which the ESP8266 Station needs to be connected.
Response	+CWJAP:<ssid>,<bssid>,<channel>,<rssi>  OK	OK or +CWJAP:<error code>  ERROR
Parameters	<ssid>: a string parameter showing the SSID of the target AP.	<ul style="list-style-type: none"><li>• &lt;ssid&gt;: the SSID of the target AP.</li><li>• &lt;pwd&gt;: password, MAX: 64-byte ASCII.</li><li>• [&lt;bssid&gt;]: the target AP's MAC address, used when multiple APs have the same SSID.</li><li>• &lt;error code&gt;: (for reference only)<ul style="list-style-type: none"><li>▶ 1: connection timeout.</li><li>▶ 2: wrong password.</li><li>▶ 3: cannot find the target AP.</li><li>▶ 4: connection failed.</li></ul></li></ul> <p>This command requires Station mode to be active. Escape character syntax is needed if SSID or password contains any special characters, such as , or " or \.</p>
Note	The configuration changes will be saved in the system parameter area in the flash.	
Examples	<p>AT+CWJAP="abc", "0123456789"</p> <p>For example, if the target AP's SSID is "ab\,c" and the password is "0123456789\", the command is as follows:</p> <p>AT+CWJAP="ab\\,c", "0123456789\\\""</p> <p>If multiple APs have the same SSID as "abc", the target AP can be found by BSSID:</p> <p>AT+CWJAP="abc", "0123456789", "ca:d7:19:d8:a6:44"</p>	

# TCP Commands

Command	Description
AT+CIPSTATUS	Gets the connection status
AT+CIPDOMAIN	DNS function
AT+CIPSTART	Establishes TCP connection, UDP transmission or SSL connection
AT+CIPSSLSIZE	Sets the size of SSL buffer
AT+CIPSEND	Sends data
AT+CIPSENDEX	Sends data when length of data is <length>, or when \0 appears in the data
AT+CIPSENDERBUF	Writes data into TCP-send-buffer
AT+CIPBUFRESET	Resets the segment ID count
AT+CIPBUFSTATUS	Checks the status of TCP-send-buffer
AT+CIPCHECKSEQ	Checks if a specific segment is sent or not
AT+CIPCLOSE	Closes TCP/UDP/SSL connection
AT+CIFSR	Gets the local IP address
AT+CIPMUX	Configures the multiple connections mode

# TCP Commands

AT+CIPSERVER	Deletes/Creates a TCP server
AT+CIPMODE	Configures the transmission mode
AT+SAVETRANSLINK	Saves the transparent transmission link in the flash
AT+CIPSTO	Sets timeout when ESP8266 runs as TCP server
AT+PING	Ping packets
AT+CIUPDATE	Upgrades the software through network
AT+CIPDINFO	Shows remote IP and remote port with +IPD
AT+CIPSNTPCFG	Configures the time domain and SNTP server.
AT+CIPSNTPTIME	Queries the SNTP time.
AT+CIPDNS_CUR	Sets user-defined DNS servers; configuration not saved in the flash
AT+CIPDNS_DEF	Sets user-defined DNS servers; configuration saved in the flash

# IoT - Thingspeak.com

← → ↻ 🔒 Secure | https://thingspeak.com/channels/new

Apps AVR Fuse Calculator IoT Analytics - Thing Talon - aThemes doc The Alpha M. Diet P Ultra Home Slack Ultra Home - Asana The Alpha M. Week EdCast | Metasploit E PyQ/PySide tut Dr@g's Places map phpmymadmin hack Other bookmark

ThingSpeak™ Channels Apps Community Support How to Buy Account Sign Out

New Channel

Name

Temperature Logger

Description

Field 1

Temperature

☒

Field 2

☐

Field 3

☐

Field 4

☐

Field 5

☐

Field 6

☐

Field 7

☐

Field 8

☐

Metadata

Tags

(Tags are comma separated)

Link to External Site

http://

Elevation

Show Location

☐

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Latitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the latitude of the city of London is 51.5072.
- Longitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
- Elevation:** Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.

Using the Channel

You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then visualize data and transform it using [ThingSpeak Apps](#).

See [Tutorial: ThingSpeak and MATLAB](#) for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

[Learn More](#)

[Download file from this page](#)

# IoT - Thingspeak.com

Secure | <https://thingspeak.com/channels/new>

Apps AVR Fuse Calculator IoT Analytics - Thing Talon - aThemes doc The Alpha M. Diet Pi Ultra Home Slack Ultra Home - Asana The Alpha M. Weekly EdCast | Metasploit PyQt/PySide tut Dr@g's Places map phpmyadmin hack Other bookmarks

ThingSpeak™ Channels Apps Community Support How to Buy Account Sign Out

Field 6

Field 7

Field 8

Metadata

Tags

(Tags are comma separated)

Link to External Site

Elevation

Show Location ☐

Latitude

Longitude

Show Video ☐

☒ YouTube

☐ Vimeo

Video URL

Show Status ☐

[Save Channel](#)

- **Longitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
- **Elevation:** Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.

## Using the Channel

You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then visualize data and transform it using [ThingSpeak Apps](#).

See [Tutorial: ThingSpeak and MATLAB](#) for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

[Learn More](#)



# IoT - Thingspeak.com

## Update a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=D4SUXF8GPW96TGD8&field
```

Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

### Write API Key

Key

[Generate New Write API Key](#)

### Read API Keys

Key

Note

[Save Note](#) [Delete API Key](#)

[Generate New Read API Key](#)

### Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

### API Keys Settings

- Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click [Generate New Write API Key](#).
- Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click [Generate New Read API Key](#) to generate an additional read key for the channel.
- Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

### API Requests

#### Update a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=D4SUXF8GPW96TGD8&field
```

#### Get a Channel Feed

```
GET https://api.thingspeak.com/channels/330528/feeds.json?api_key=9
```

#### Get a Channel Field

```
GET https://api.thingspeak.com/channels/330528/fields/1.json?api_key
```

#### Get Channel Status Updates

```
GET https://api.thingspeak.com/channels/330528/status.json?api_key=5
```

[Learn More](#)

[Download file from this page](#)



# Get Request to Push Data

## Update a Channel Feed

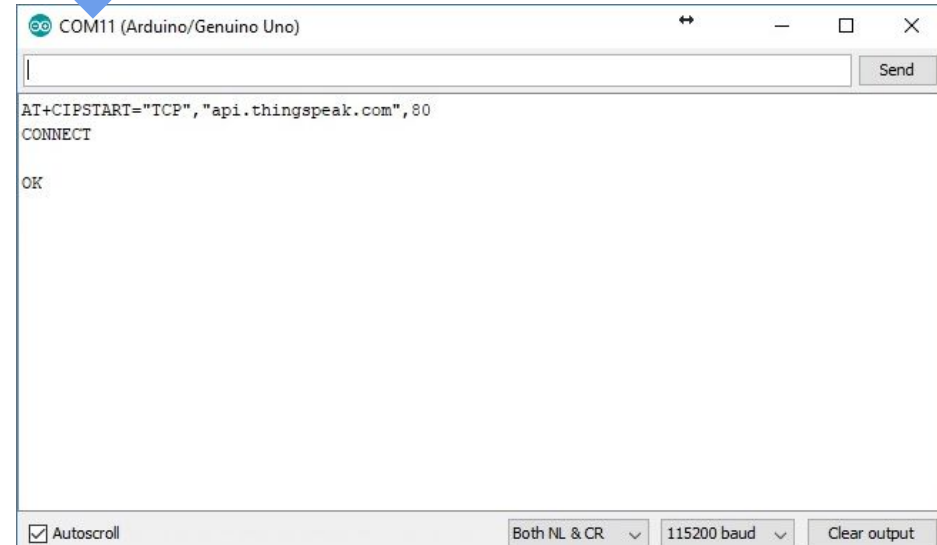
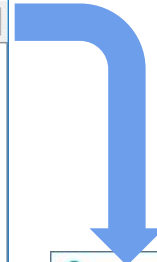
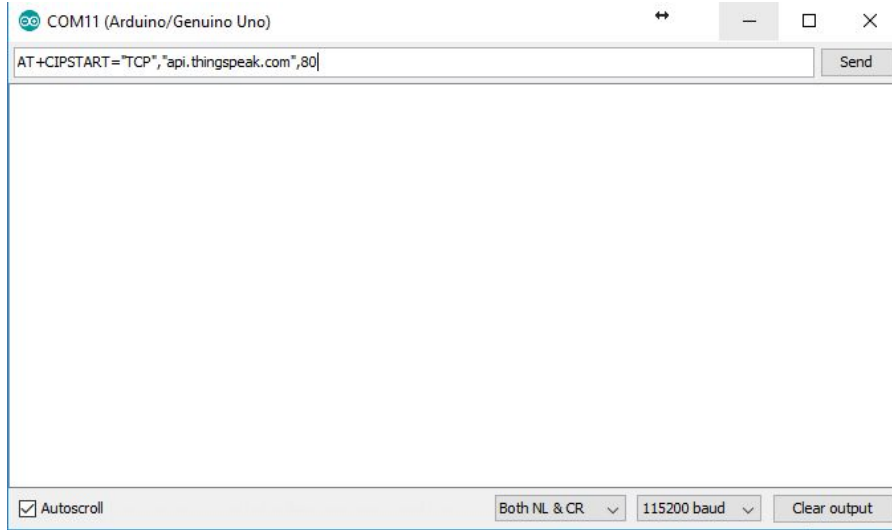
```
GET https://api.thingspeak.com/update?api_key=D4SUXF8GPW96TGD8&field1=9
```

```
1 AT+CIPSTART="TCP","api.thingspeak.com",80
2 AT+CIPSEND=73
3 GET https://api.thingspeak.com/update?api_key=D4SUXF8GPW96TGD8&field1=9
4
```

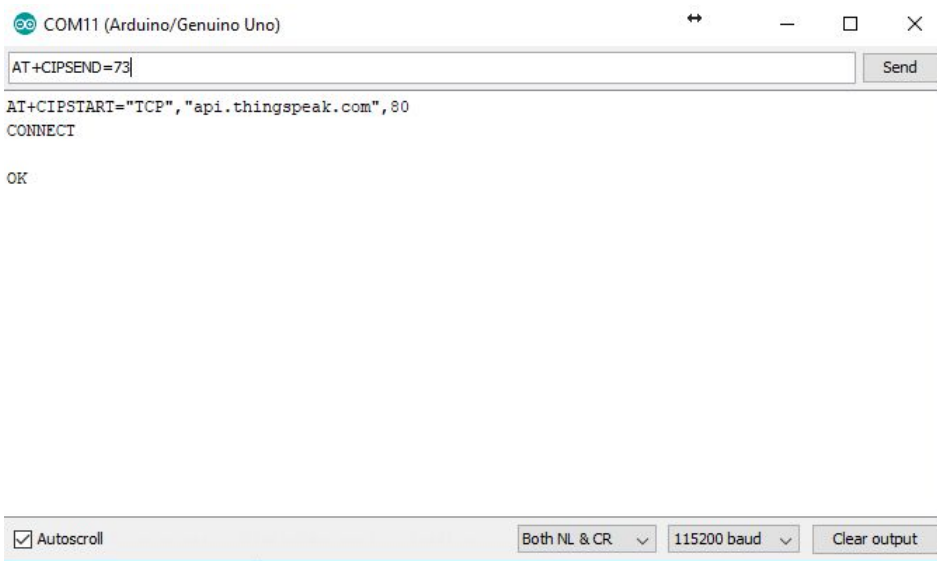
Total number of Bytes in the  
GET request. Including CR and  
LF at the end

Value to  
be pushed

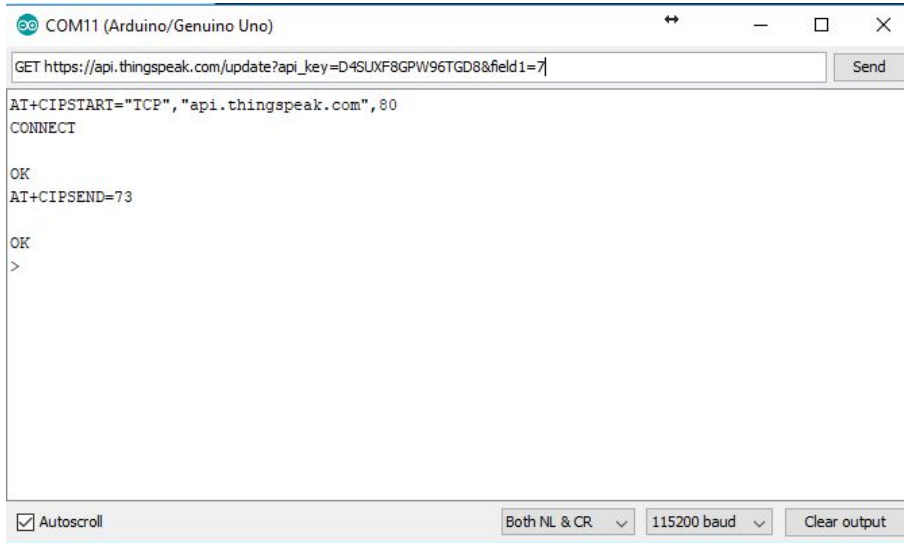
# Push Data from Serial Monitor - TCP connection



# Push Data from Serial Monitor - Specify number of bytes to be sent



# Push Data from Serial Monitor - send the get request



A screenshot of a serial monitor window titled "COM11 (Arduino/Genuino Uno)". The input field contains the command: `GET https://api.thingspeak.com/update?api_key=D4SUXF8GPW96TGD8&field1=7`. A blue arrow points from the "Send" button of this window to the right window. The output area shows the following text: `AT+CIPSTART="TCP", "api.thingspeak.com", 80`, `CONNECT`, `OK`, `AT+CIPSEND=73`, `OK`, and `>`. At the bottom, there are checkboxes for "Autoscroll", dropdown menus for "Both NL & CR" and "115200 baud", and a "Clear output" button.

```
COM11 (Arduino/Genuino Uno)
```

GET https://api.thingspeak.com/update?api\_key=D4SUXF8GPW96TGD8&field1=7

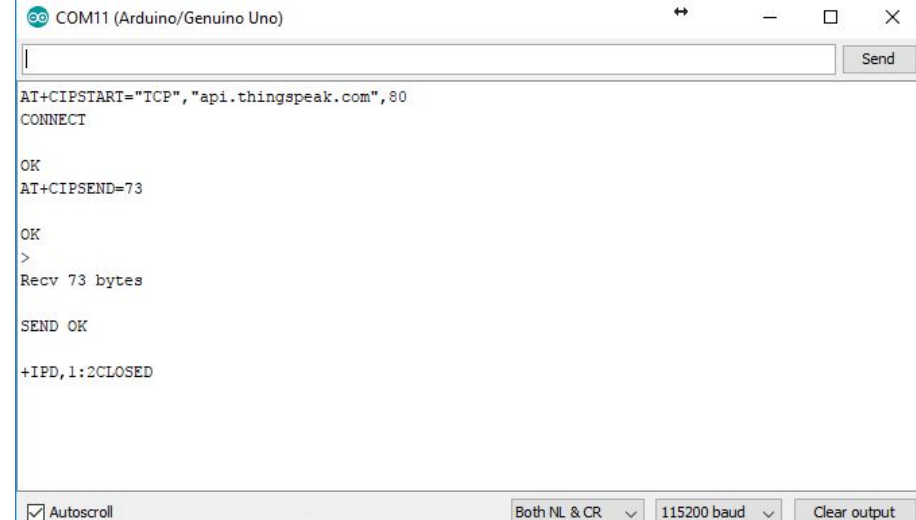
Send

```
AT+CIPSTART="TCP", "api.thingspeak.com", 80
CONNECT

OK
AT+CIPSEND=73

OK
>
```

☒ Autoscroll    Both NL & CR    115200 baud    Clear output



A screenshot of a serial monitor window titled "COM11 (Arduino/Genuino Uno)". The output area shows the following text: `AT+CIPSTART="TCP", "api.thingspeak.com", 80`, `CONNECT`, `OK`, `AT+CIPSEND=73`, `OK`, `>`, `Recv 73 bytes`, `SEND OK`, and `+IPD,1:2CLOSED`. At the bottom, there are checkboxes for "Autoscroll", dropdown menus for "Both NL & CR" and "115200 baud", and a "Clear output" button.

```
COM11 (Arduino/Genuino Uno)
```

AT+CIPSTART="TCP", "api.thingspeak.com", 80

CONNECT

OK

AT+CIPSEND=73

OK

>

Recv 73 bytes

SEND OK

+IPD,1:2CLOSED

☒ Autoscroll    Both NL & CR    115200 baud    Clear output

# Visualize the data

## Temperature Logger

Channel ID: 330528

Author: anahera

Access: Private

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

Data Export

MATLAB Analysis

MATLAB Visualization

## Channel Stats

Created: about an hour ago

Updated: 3 minutes ago

Last entry: 3 minutes ago

Entries: 3

