



instructables

## Getting Time From Internet Using ESP8266 | NTP Clock Project With ESP8266 Nodemcu



by Utsource

In this tutorial we'll see how to get time using the ESP8266/nodemcu with Arduino IDE. Getting time is especially useful in data logging to timestamp your readings. If your ESP8266 project has access to the Internet, you can get time using Network Time Protocol (NTP) – you don't need any additional hardware.

You can connect your ESP8266 to your wifi network and it will be a clock which will be synchronized with network, so if once you Uploaded the code it will get time from internet so it will always display correct time.

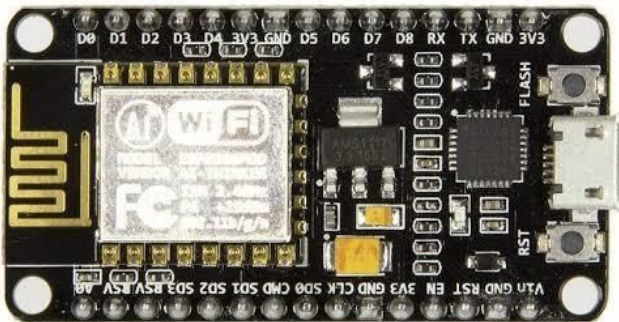


## Step 1: Things You Need

For this project you'll need very few things :

ESP8266/NODEMCU

A USB cable to program it.



## Step 2: What Is a NTP and How It Will Work?

What is a NTP : An NTP stands for Network Time Protocol. It's a standard Internet Protocol (IP) for synchronizing the computer clocks to some reference over a network.

The protocol can be used to synchronize all networked devices to Coordinated Universal Time (UTC).

NTP sets the clocks of computers to UTC, any local time zone offset or day light saving time offset is applied by the client. In this manner clients can synchronize to servers regardless of location and time zone differences.

The client device such as ESP8266 connects to the server using the User Datagram Protocol (UDP) on port 123.

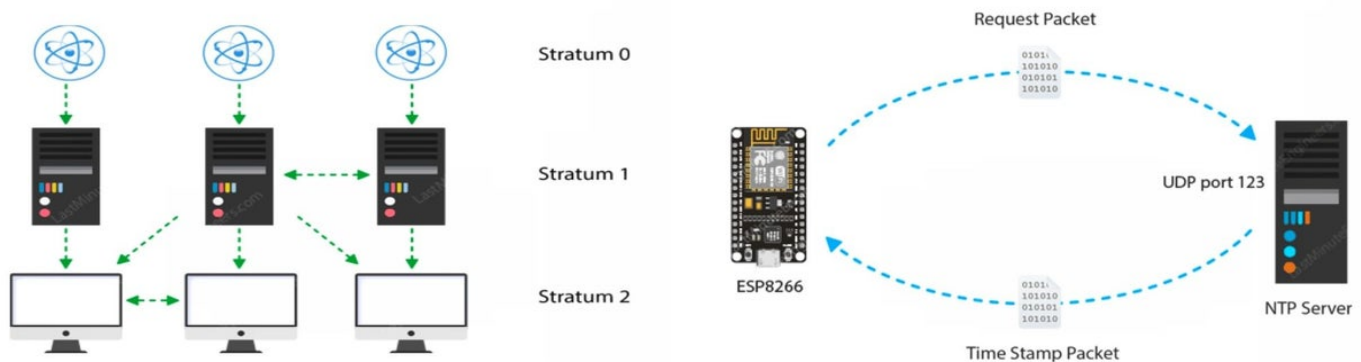
A client then transmits a request packet to a NTP server.

In response to this request the NTP server sends a time stamp packet.

A time stamp packet contains multiple information like UNIX timestamp, accuracy, delay or timezone.

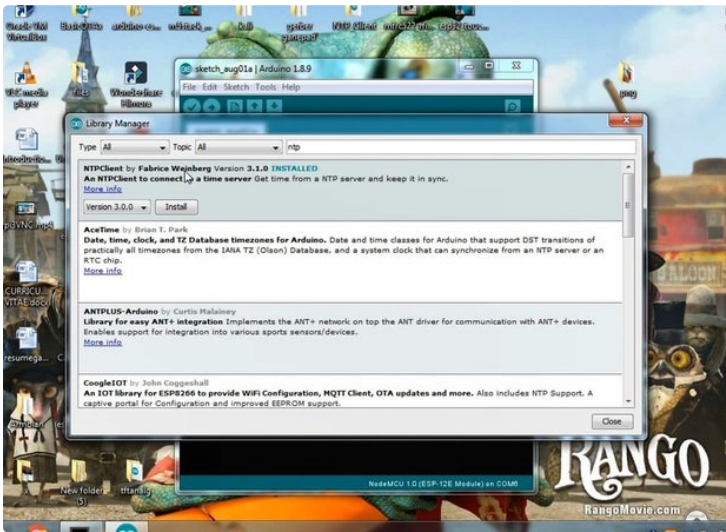
A client can then parse out current date & time values.

How it will work for us :



## Step 3: Installing Library on Arduino IDE

In your Arduino IDE go to Libraries manager and search for NTP and just download the NTP client library as i downloaded, refer image for further help.



## Step 4: Coding Part

Please copy the following code & put your network credentials in your code then

You need to set offsettime for me it is 19800

Because my timezone is utc+5:30 so

UTC +5:30=5.5\*60\*60=19800

UTC+1=1\*60\*60=3600

CALCULATE your timezone and edit it and then upload the code.

```
#include "NTPClient.h"
```

```
#include "ESP8266WiFi.h"
```

```
#include "WiFiUdp.h"
```

```
const char *ssid = "*****";
```

```
WiFi.begin(ssid, password);
```

```
while ( WiFi.status() != WL_CONNECTED ) {
  delay ( 500 );
  Serial.print ( "." );
}
```

```
timeClient.begin();
}
```

```
void loop() {
  timeClient.update();
}
```

```
const char *password = "*****";
```

```
const long utcOffsetInSeconds = 19800;
```

```
char daysOfTheWeek[7][12] = {"Sunday", "Monday",
  "Tuesday", "Wednesday", "Thursday", "Friday",
  "Saturday"};
```

```
// Define NTP Client to get time
```

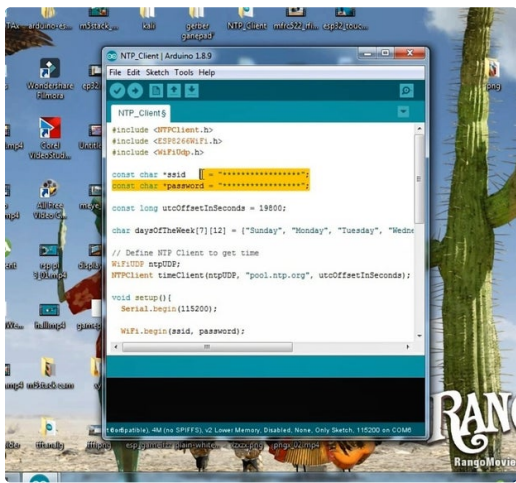
```
WiFiUDP ntpUDP;
```

```
NTPClient timeClient(ntpUDP, "pool.ntp.org",
  utcOffsetInSeconds);
```

```
void setup(){
  Serial.begin(115200);
```

```
  Serial.print(daysOfTheWeek[timeClient.getDay()]);
  Serial.print(", ");
  Serial.print(timeClient.getHours());
  Serial.print(":");
  Serial.print(timeClient.getMinutes());
  Serial.print(":");
  Serial.println(timeClient.getSeconds());
  //Serial.println(timeClient.getFormattedTime());
```

```
  delay(1000);
}
```



$$\text{UTC}+5:30 = 5.5 \times 60 \times 60 = 19800$$

$$\text{UTC}+1 = 1 \times 60 \times 60 = 3600$$

## Step 5: Getting TIME

After Uploading the code to Esp8266 you can open the serial monitor and if everything is good then you will able to get the time on the serial monitor as i am able to get the time in my serial monitor. So with this project you can attach any display and make it a proper network clock. So have fun making your network clock.

