Getting Date & Time From NTP Server With ESP32

Jeder einmal in eine Weile, werden Sie kommen über eine Idee, wo die Zeit ein zentrales Anliegen. Stellen Sie sich beispielsweise ein relais, das aktiviert werden soll, zu einer bestimmten Zeit oder ein Daten-logger, die zum speichern von Werten in präzisen Intervallen.

Das erste, was kommt in Ihre Geist ist die Verwendung einer RTC (Real Time Clock) - chip. Aber diese chips sind nicht ganz richtig so, die Sie benötigen, um manuelle Anpassungen über und über wieder zu synchronisieren.

Die Lösung hier ist die Verwendung von **Network Time Protocol** (NTP). Wenn Ihr ESP32 Projekt hat Zugriff auf das Internet, können Sie Datum und Zeit (mit einer Genauigkeit innerhalb von ein paar Millisekunden (UTC), die für **FREIES** . Sie brauchen keine zusätzliche hardware.

Was ist ein NTP?

Ein NTP steht für Network Time Protocol . Es ist ein standard-Internet-Protokoll (IP) zum synchronisieren von die computer-Uhren, um einige Referenz über ein Netzwerk.

Das Protokoll kann verwendet werden, um synchronisieren Sie alle Ihre vernetzten Geräte zu **Coordinated Universal Time** (UTC) innerhalb von ein paar Millisekunden (50 Millisekunden, die über das öffentliche Internet und unter 5 Millisekunden in einer LAN-Umgebung).

UTC (Universal Time Coordinated), ist eine weltweite zeitstandard, die eng mit GMT (Greenwich Mean Time). UTC ändert sich nicht, es ist das gleiche weltweit.

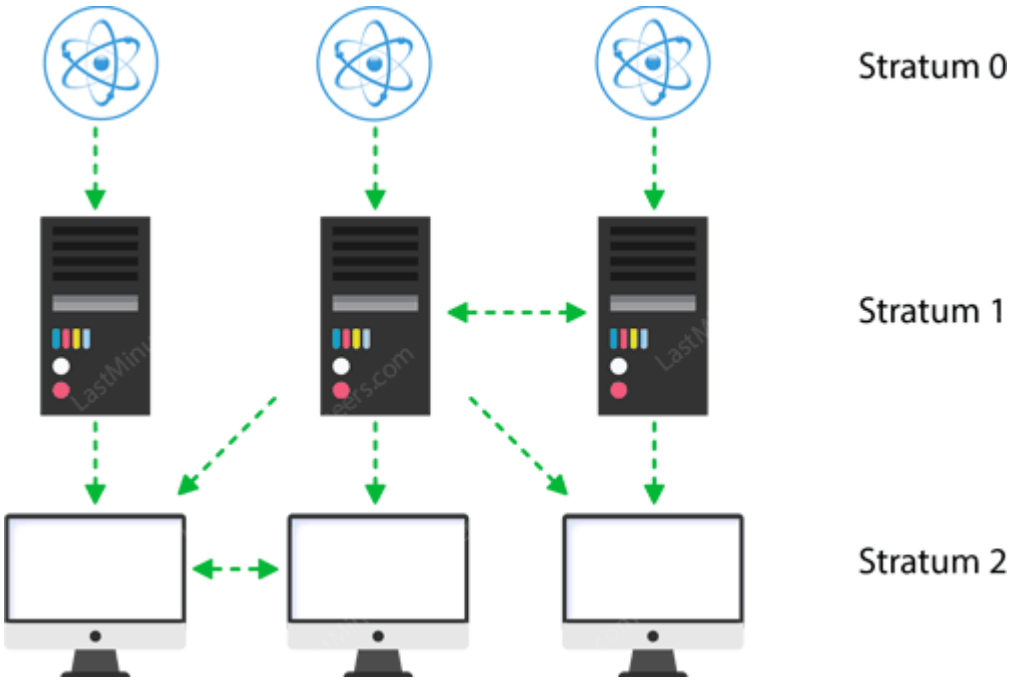
NTP stellt die Uhren von Computern in UTC, die alle lokalen Zeitzonen-offset oder Tag Licht sparen Zeit-offset wird angewendet, indem der Kunde. Auf diese Weise können die clients synchronisieren zu Servern, unabhängig von Ort und Zeit zone Unterschiede.

NTP-Architektur

NTP nutzt ein hierarchisches Architektur. Jede Ebene in der Hierarchie ist bekannt als eine **Schicht** .

Ganz oben sind high-Präzision des Uhrwerks Geräte, wie Atomuhren, GPS-Funkuhren, bekannt als stratum 0 hardware-Uhren.

Stratum-1-Server eine direkte Verbindung zu einem stratum 0 hardware-Uhr und haben daher die genaue Zeit.

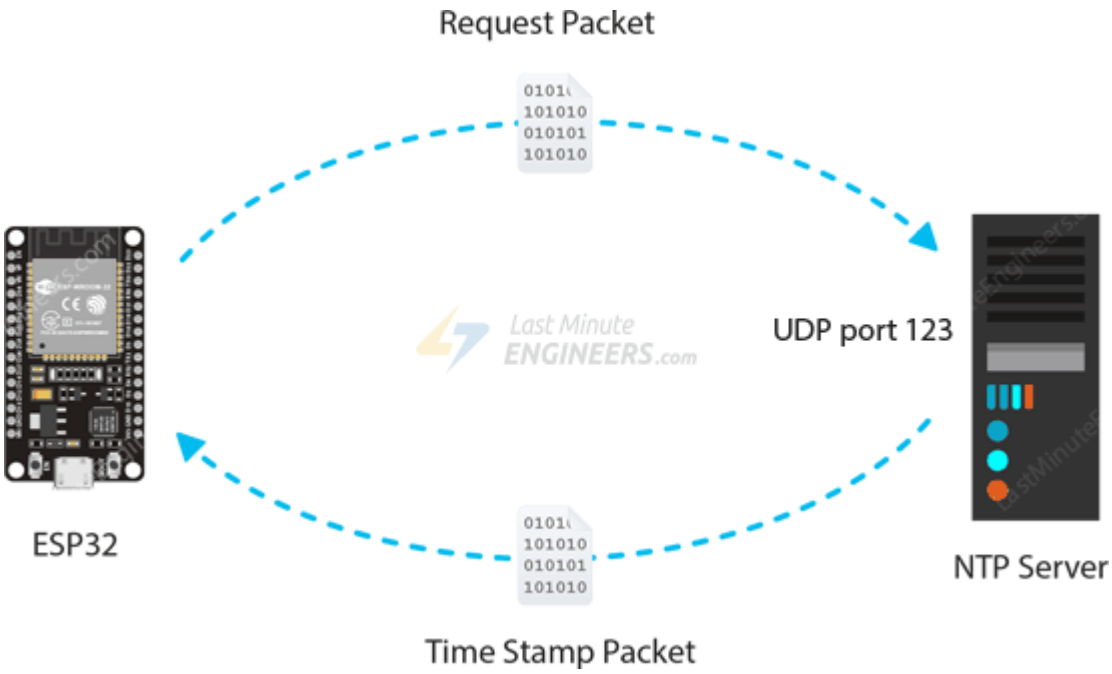


Jede Schicht in der Hierarchie synchronisiert mit die Schicht oben und fungieren als Server für die untere Schicht Computern.

How NTP Works?

NTP kann in betrieben in einer Reihe von Möglichkeiten. Die häufigste Konfiguration ist zu **bedienen Sie in der client-server-Modus** . Die grundlegende Arbeits Prinzip ist wie folgt:

1. Das client-Gerät wie dem ESP32 Verbindung zum server mit dem User Datagram Protocol (UDP) an port 123.
2. Ein client sendet ein request-Paket an einen NTP-server.
3. In Reaktion auf diese Forderung der NTP-server sendet einen Zeitstempel Paket.
4. Ein Zeitstempel-Paket enthält mehrere Informationen wie der UNIX-timestamp, Genauigkeit, Verzögerung oder Zeitzone.
5. Ein client kann dann analysiert aktuelle Datum & Zeit-Werten.



Vorbereitung der Arduino IDE

Genug der Theorie, Gehen wir mal Praktisch!

Aber bevor Sie sich weiter in diesem tutorial sollten Sie das ESP32-add-on installiert ist in Ihrem Arduino-IDE. Folgen Sie den unten Tutorials bereiten Sie Ihre Arduino IDE zu arbeiten mit die ESP32, wenn Sie nicht bereits getan haben.

Einblick In ESP32 Funktionen & Verwendung Mit der Arduino IDE
Paar Jahre zurück, ESP8266 nahm die embedded-IoT-Welt durch Sturm. Für weniger als $3, Sie erhalten könnte ein programmierbarer WLAN-fähigen mikrocontroller zu können...

Erste Datum und die Uhrzeit über NTP-Server

Die folgende Skizze wird Ihnen vollständig zu verstehen, wie um Datum und Uhrzeit vom NTP-Server.

Bevor Sie den Kopf für das hochladen der Skizze, Sie müssen **machen einige änderungen** , um es für Sie arbeiten.

- Sie ändern müssen die folgenden zwei Variablen mit Ihren Netzwerk-Anmeldeinformationen, so dass ESP32 kann eine Verbindung mit dem bestehenden Netzwerk.

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

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

- Sie benötigen zum einstellen der UTC-offset für Ihre Zeitzone in Millisekunden. Finden Sie die Liste der UTC-Zeit-offsets . Hier sind einige Beispiele für die verschiedenen Zeitzonen:
  - Für UTC -5.00 : -5 * 60 * 60 : -18000
  - Für UTC +1.00 : 1 * 60 * 60 : 3600
  - Für UTC +0.00 : 0 * 60 * 60 : 0

```
const long  gmtOffset_sec = 3600;
```

- Ändern der Sommerzeit-offset, in Millisekunden. Wenn Ihr Land beobachtet die Sommerzeit einstellen, dass es 3600. Andernfalls, setzen Sie es auf 0.

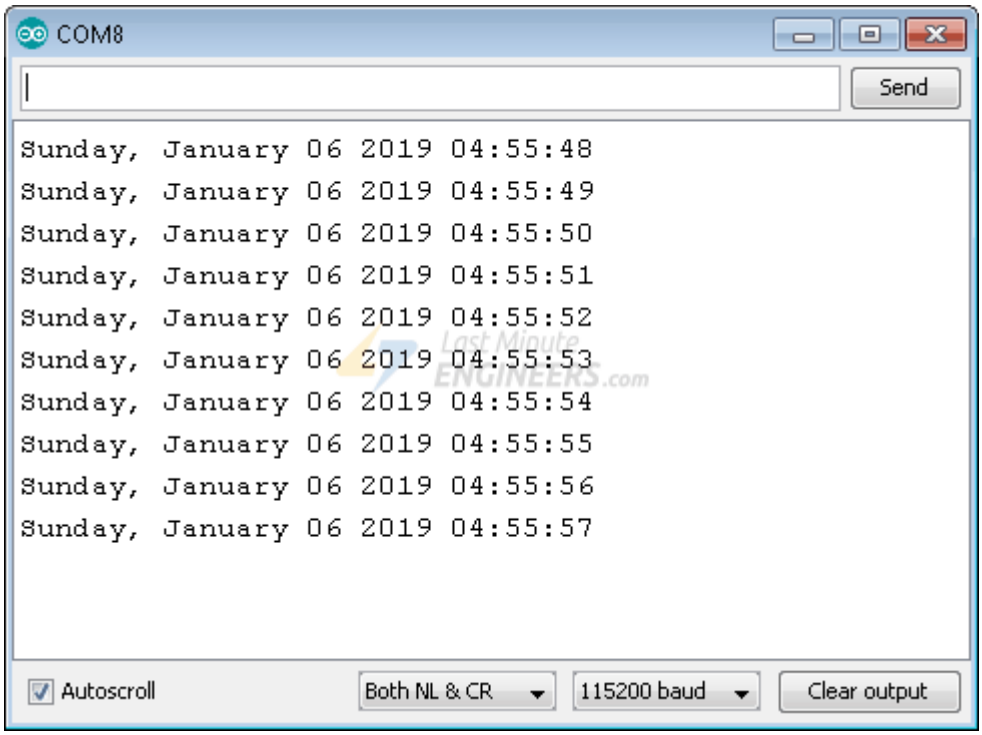
```
const int   daylightOffset_sec = 3600;
```

Sobald Sie fertig sind, gehen Sie voran und versuchen Sie die Skizze aus.

```
#include <WiFi.h>
#include "time.h"

const char* ssid       = "YOUR_SSID";
const char* password   = "YOUR_PASS";

const char* ntpServer = "pool.ntp.org";
const long  gmtOffset_sec = 3600;
const int   daylightOffset_sec = 3600;

void printLocalTime()
{
  struct tm timeinfo;
  if(!getLocalTime(&timeinfo)){
    Serial.println("Failed to obtain time");
    return;
  }
  Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");
}

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

  //connect to WiFi
  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
  }
  Serial.println(" CONNECTED");

  //init and get the time
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  printLocalTime();

  //disconnect WiFi as it's no longer needed
  WiFi.disconnect(true);
  WiFi.mode(WIFI_OFF);
}

void loop()
{
```

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

Nach dem hochladen der Skizze, drücken Sie die EN-Schaltfläche auf Ihrem ESP32, und Sie sollten das Datum und die Zeit jede Sekunde, wie unten gezeigt.



Code Erklärung

Werfen wir einen kurzen Blick auf den code werfen, um zu sehen, wie es funktioniert. Erste, wir sind die Bibliotheken, die benötigt werden für dieses Projekt.

- **WiFi.h** - Bibliothek bietet ESP32 bestimmten WiFi-Methoden, die wir aufrufen, um eine Verbindung zum Netzwerk.
- **Zeit.h** ist der ESP32 native Zeit-Bibliothek-die nicht anmutig NTP-server-Synchronisation.

```
#include <WiFi.h>
#include "time.h"
```

Neben, wir set up ein paar Konstanten wie SSID, WLAN-Passwort, Passwort UTC Offset & Sommerzeit-offset, die Sie bereits kennen.

Zusammen mit, dass wir angeben müssen, die Adresse des NTP-Servers ein, die wir nutzen wollen. **pool.ntp.org** ist ein open-NTP-Projekt-ideal für Dinge wie diese.

```
const char* ntpServer = "pool.ntp.org";
```

Die pool.ntp.org automatische Erfassung Zeit-Server, die geografisch nahe für Sie. Aber wenn Sie möchten, wählen Sie explizit, verwenden Sie eine der **sub-Zonen** der pool.ntp.org.

Bereich	HostName
Weltweit	pool.ntp.org
Asien	asia.pool.ntp.org

Europa	europe.pool.ntp.org
North America	north-america.pool.ntp.org
Ozeanien	oceania.pool.ntp.org
South America	south-america.pool.ntp.org

Im setup-Abschnitt, wir erste initialisiert die serielle Kommunikation mit PC und verbinden Sie das WiFi-Netzwerk mit `WiFi.begin()` Funktion.

```
Serial.begin(115200);

//connect to WiFi
Serial.printf("Connecting to %s ", ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println(" CONNECTED");
```

Sobald ESP32 mit dem Netzwerk verbunden ist, initialisieren wir den NTP-client mit `configTime()` Funktion, um Datum und Uhrzeit von einem NTP-server.

```
//init and get the time
configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
```

Jetzt können wir rufen Sie das `printLocalTime()` benutzerdefinierte Funktion, Wann immer wir wollen zu drucken current date & time.

`getLocalTime()` Funktion ist verwendet zu übertragen-request-Paket an einen NTP-server und analysieren die empfangenen Zeitstempel Paket in einem maschinenlesbaren format. Es braucht Zeit-Struktur als parameter.

Sie können auf die Datums - & Zeit-Informationen durch den Zugriff auf die Mitglieder dieser Struktur.

%A	gibt den Tag der Woche
%B	gibt den Monat des Jahres
%d	gibt den Tag des Monats
%Y	returns Jahr
%H	Rückgabe Stunde
%M	returns Minuten
%S	returns Sekunden

```
void printLocalTime()
{
  struct tm timeinfo;
  if(!getLocalTime(&timeinfo)){
    Serial.println("Failed to obtain time");
    return;
  }
  Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");
}
```