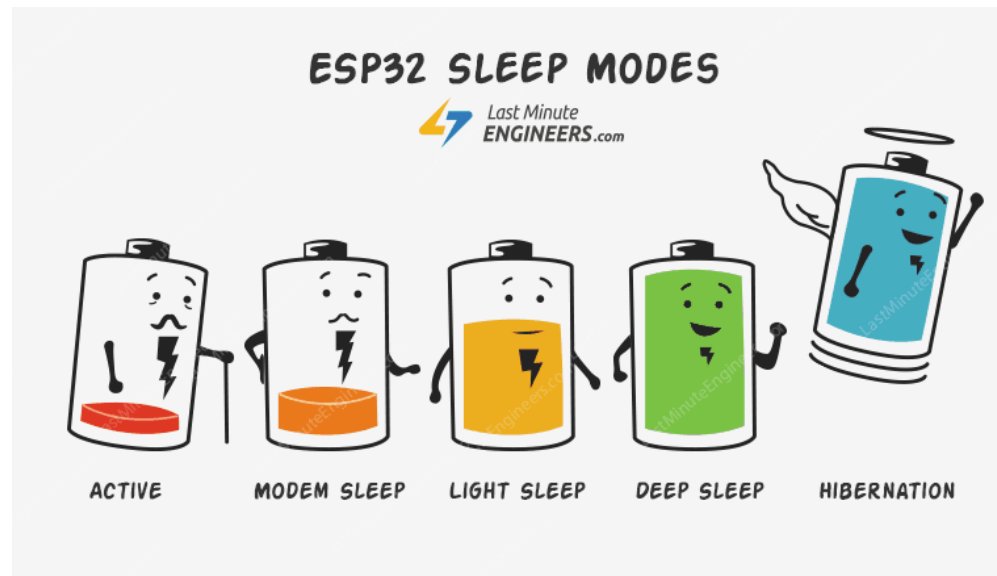


# ESP32 Low-Power režimi i merenje vremena



# Interna struktura ESP32

- ESP32 je mikrokontrolerska platforma bogata resursima koji podržavaju širok spektar IoT aplikacija:
  - Dual-Core 32-bitni MCU + 448 KB ROM + 520 KB SRAM + 4MB Flash
  - WiFi modul
  - Bluetooth modul
  - Kriptografski akcelerator
  - RTC modul
  - Periferijski moduli
- Ovi resursi kada su aktivni mogu trošiti značajnu količinu energije, zbog čega su podržani različiti režimi potrošnje:
  1. Active Mode
  2. Modem Sleep Mode
  3. Light Sleep Mode
  4. Deep Sleep Mode
  5. Hibernation Mode



# ESP32 Active Mode

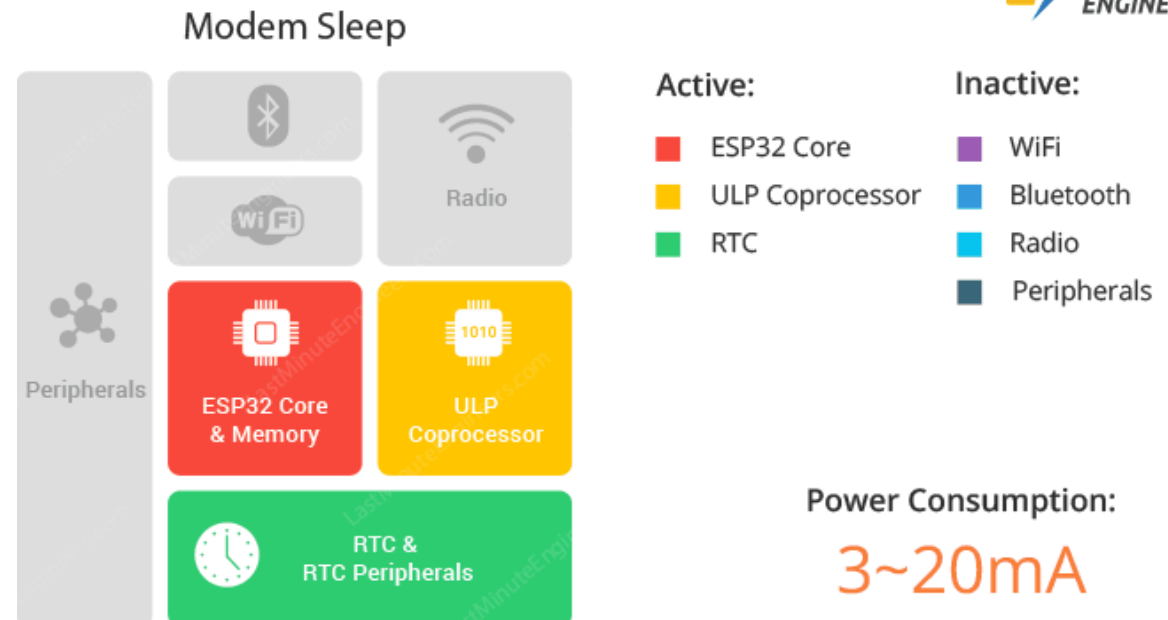
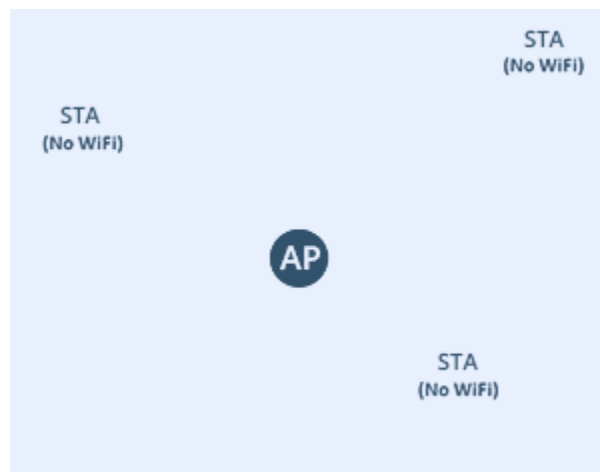
- U ovom režimu, svi interni moduli su aktivni
- Trenutna potrošnja u velikoj meri zavisi od stanja u kojem su interfejsi za bežičnu komunikaciju, prvenstveno WiFi:

Mode	Power Consumption
Wi-Fi Tx packet 13dBm~21dBm	160~260mA
Wi-Fi/BT Tx packet 0dBm	120mA
Wi-Fi/BT Rx and listening	80~90mA



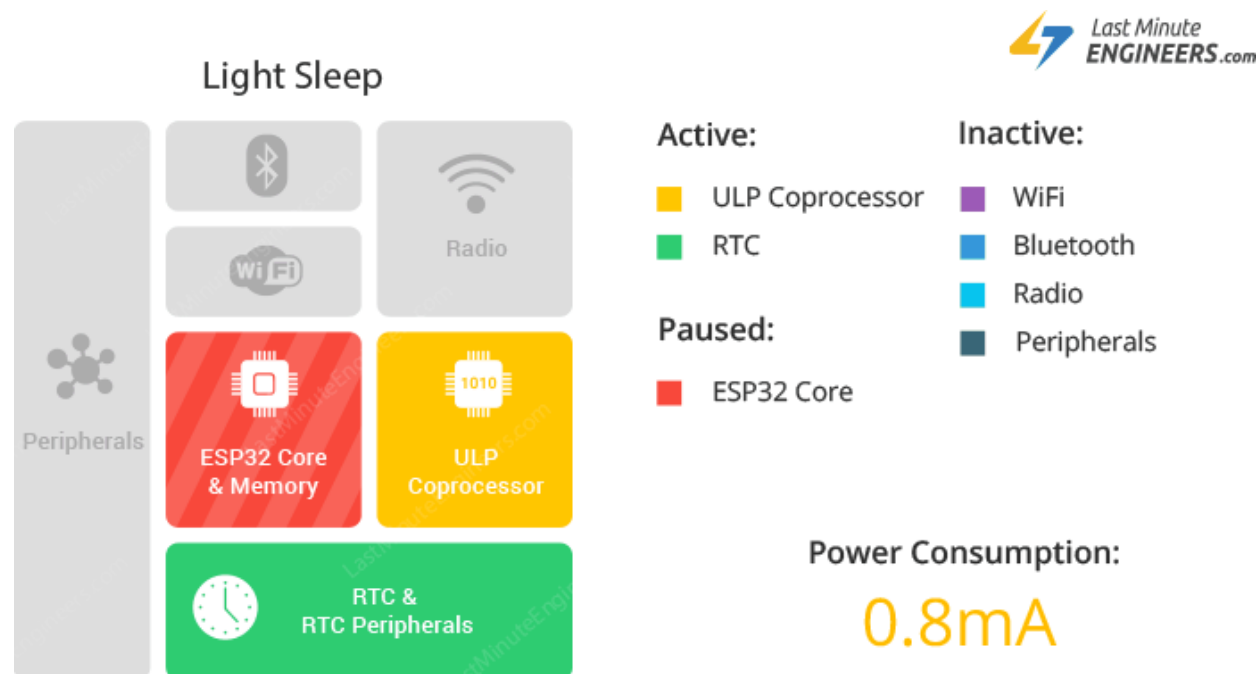
# ESP32 Modem Sleep Mode

- U ovom režimu, isključeni su WiFi, Bluetooth i radio
- U cilju održavanja konekcije sa WiFi AP, koristi se DTIM beacon mehanizam, pri čemu se bežični moduli periodično bude u skladu sa predefinisanim intervalom



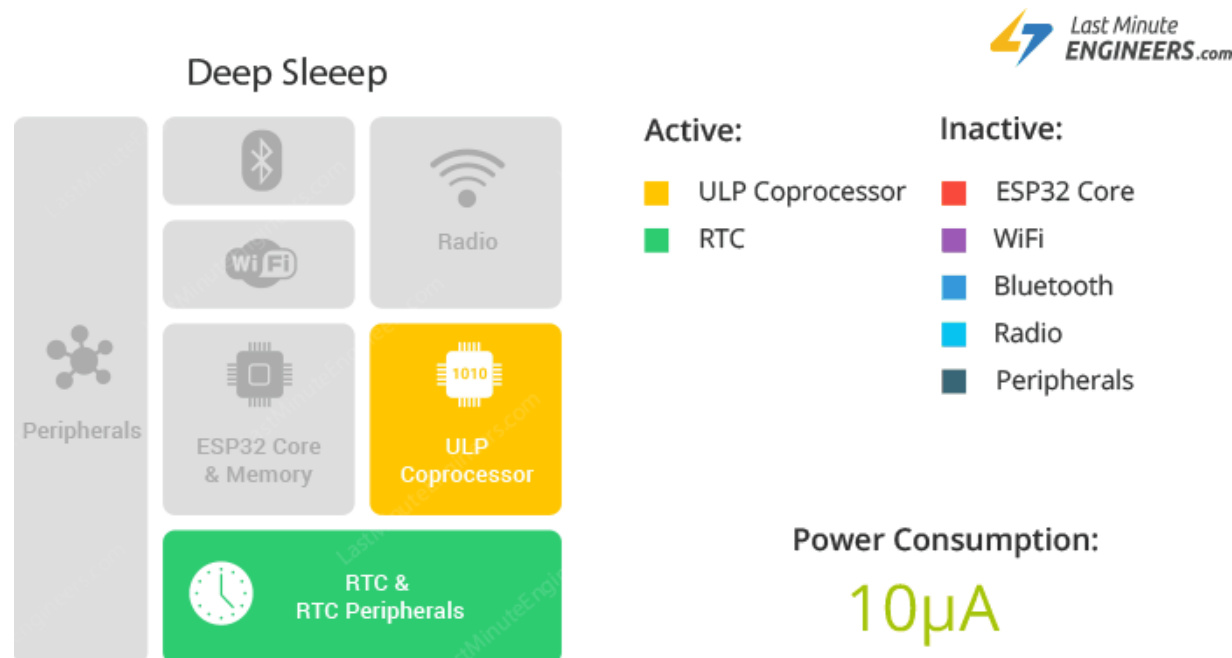
# ESP32 Light Sleep Mode

- Ovaj režim je sličan prethodnom (Modem Sleep Mode) po pitanju aktivnih periferija
- Razlika je u tome što ovaj režim koristi tzv. clock gating tehniku, koja u delovima kola sprečava propagaciju takt signala, čime svi flip-floповi zadržavaju prethodno stanje



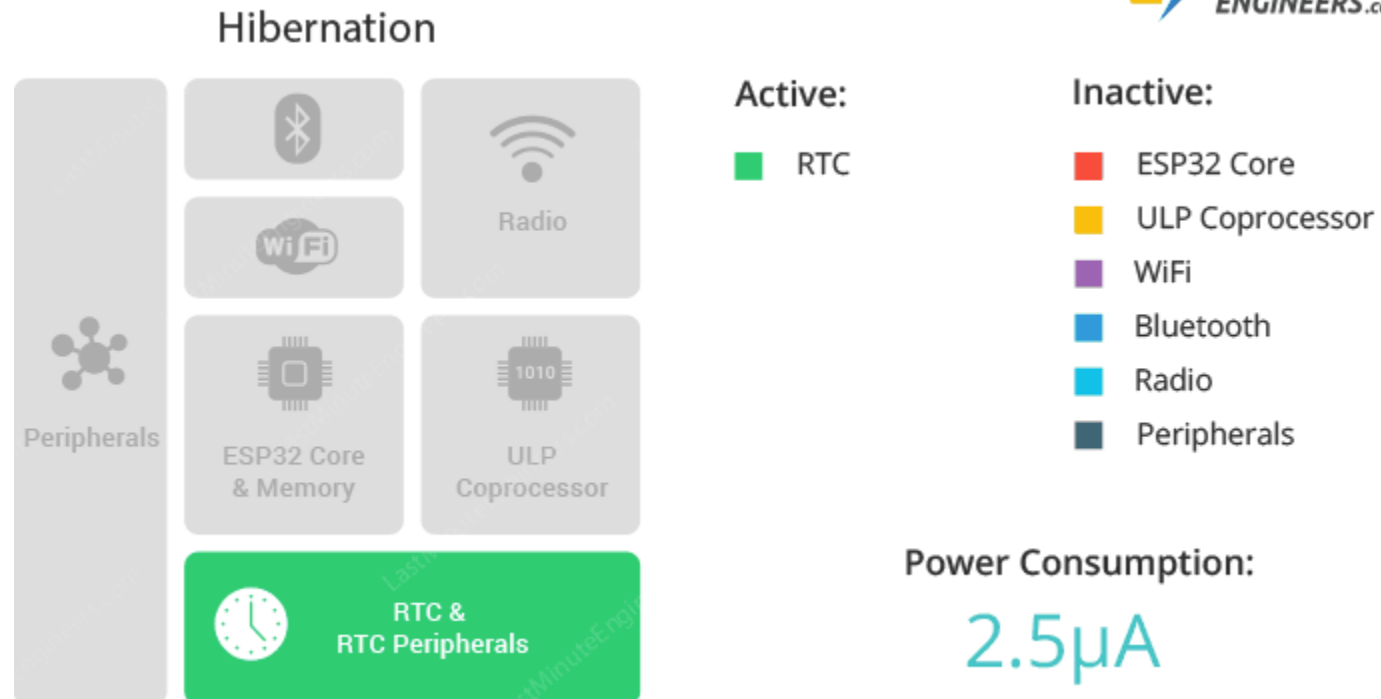
# ESP32 Deep Sleep Mode

- Svi moduli su isključeni, izuzev ULP jezgra i RTC sa pridruženim periferijama
- Potrošnja varira između 150uA (kada je ULP uključen) i 10uA
- Jedina RAM memorija koja zadržava sadržaj je ona koja pripada RTC-u, kapaciteta 8KB. Posledica: po izlasku iz ovog režima, aplikacija se resetuje



# ESP32 Hibernation Mode

- Svi moduli su isključeni, izuzev RTC sa pridruženim periferijama
- U ovom režimu se gubi sadržaj svih RAM memorija, uključujući onu koja pripada RTC-u



# Deep sleep mode i načini buđenja

- Pošto deep sleep briše sadržaj memorije, varijable koje želimo da sačuvamo moramo smestiti u RTC memoriju pomoću atributa `RTC_DATA_ATTR`, npr:

```
RTC_DATA_ATTR int bootCount = 0;
```

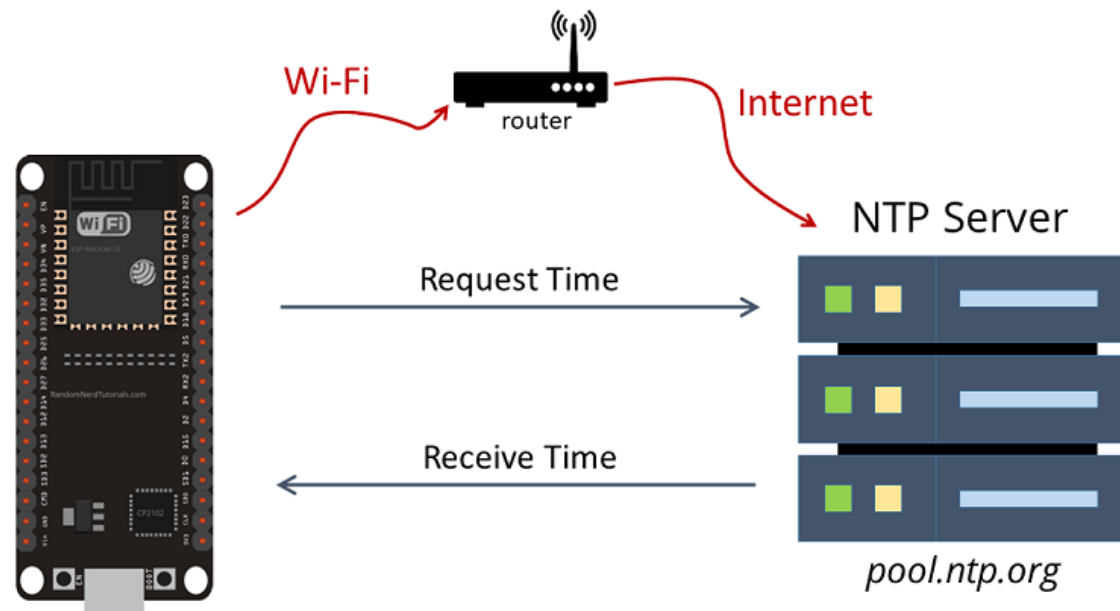
- Buđenje se može vršiti na različite načine:
  - Tajmer (RTC)
  - Touch Pad
  - Promenom stanja eksternog pina: `ext0(RTC GPIOs 0,2,4,12-15,25-27,32-39)`, ili `ext1(RTC GPIOs 32-39)`
- Primer 1: Buđenje pritiskom na taster (*Wakeup\_button.ino*)
- Primer 2: Buđenje pomoću RTC (*Wakeup\_timer.ino*)

# Merenje vremena

- UTC (Coordinated Universal Time) – univerzalno referentno vreme, koje je u datom trenutku identično, bez obzira na geografsku lokaciju
- Naše lokalno vreme je UTC+1 po zimskom, odnosno UTC+2 po letnjem računanju vremena
- [UNIX timestamp](#) je sistem za označavanje vremena, koji predstavlja broj sekundi proteklih od referentnog vremenskog trenutka (01.01.1970. 00:00:00 UTC)
- Budući da se za izražavanje Unix TS koristi 32-bitni označeni tip, do prekoračenja opsega će doći 19.01.2038. u 03:14:07 UTC
- Biblioteke za rad sa vremenom: time.h i [TimeLib.h](#)

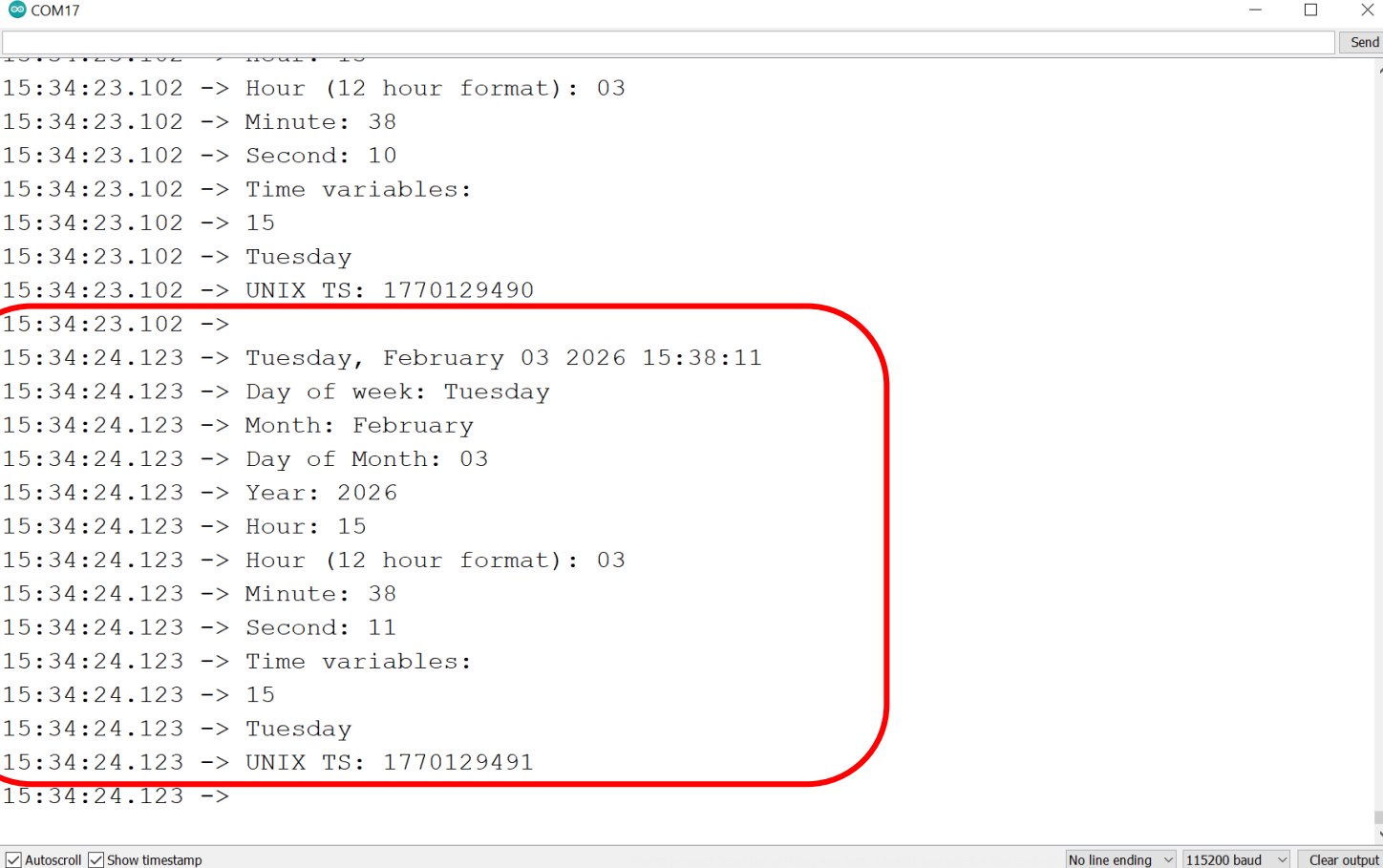
# NTP (Network Time Protocol)

- NTP je mrežni protokol koji se koristi za sinhronizaciju vremena između računarskih sistema
- Serverima kao što je [pool.ntp.org](http://pool.ntp.org) svako može proslediti zahtev i na taj način iz odgovora očitati UTC vreme u trenutku prijema zahteva
- Usled potrebe za što manjim kašnjenjem odgovora, NTP koristi UDP na transportnom sloju



# Primer 3: Sinhronizacija vremena (NTP)

- *ESP32\_date\_time.ino*



```
COM17
15:34:23.102 -> Hour (12 hour format): 03
15:34:23.102 -> Minute: 38
15:34:23.102 -> Second: 10
15:34:23.102 -> Time variables:
15:34:23.102 -> 15
15:34:23.102 -> Tuesday
15:34:23.102 -> UNIX TS: 1770129490
15:34:23.102 ->
15:34:24.123 -> Tuesday, February 03 2026 15:38:11
15:34:24.123 -> Day of week: Tuesday
15:34:24.123 -> Month: February
15:34:24.123 -> Day of Month: 03
15:34:24.123 -> Year: 2026
15:34:24.123 -> Hour: 15
15:34:24.123 -> Hour (12 hour format): 03
15:34:24.123 -> Minute: 38
15:34:24.123 -> Second: 11
15:34:24.123 -> Time variables:
15:34:24.123 -> 15
15:34:24.123 -> Tuesday
15:34:24.123 -> UNIX TS: 1770129491
15:34:24.123 ->
```

The screenshot shows a serial monitor window with a red rounded rectangle highlighting the output for the second time point (15:34:24.123). The output includes the day of the week, full date and time, day of week, month, day of month, year, hour, hour in 12-hour format, minute, second, and time variables.

# Zadatak

Napisati aplikaciju koja:

1. Sinhronizuje vreme na ESP32 pomoću NTP servera.
2. Omogućava korisniku unos vremena buđenja (sat, minut, sekund) preko serijskog porta.
3. Odlazi u Deep Sleep.
4. Budi se u trenutku koji je korisnik zadao.