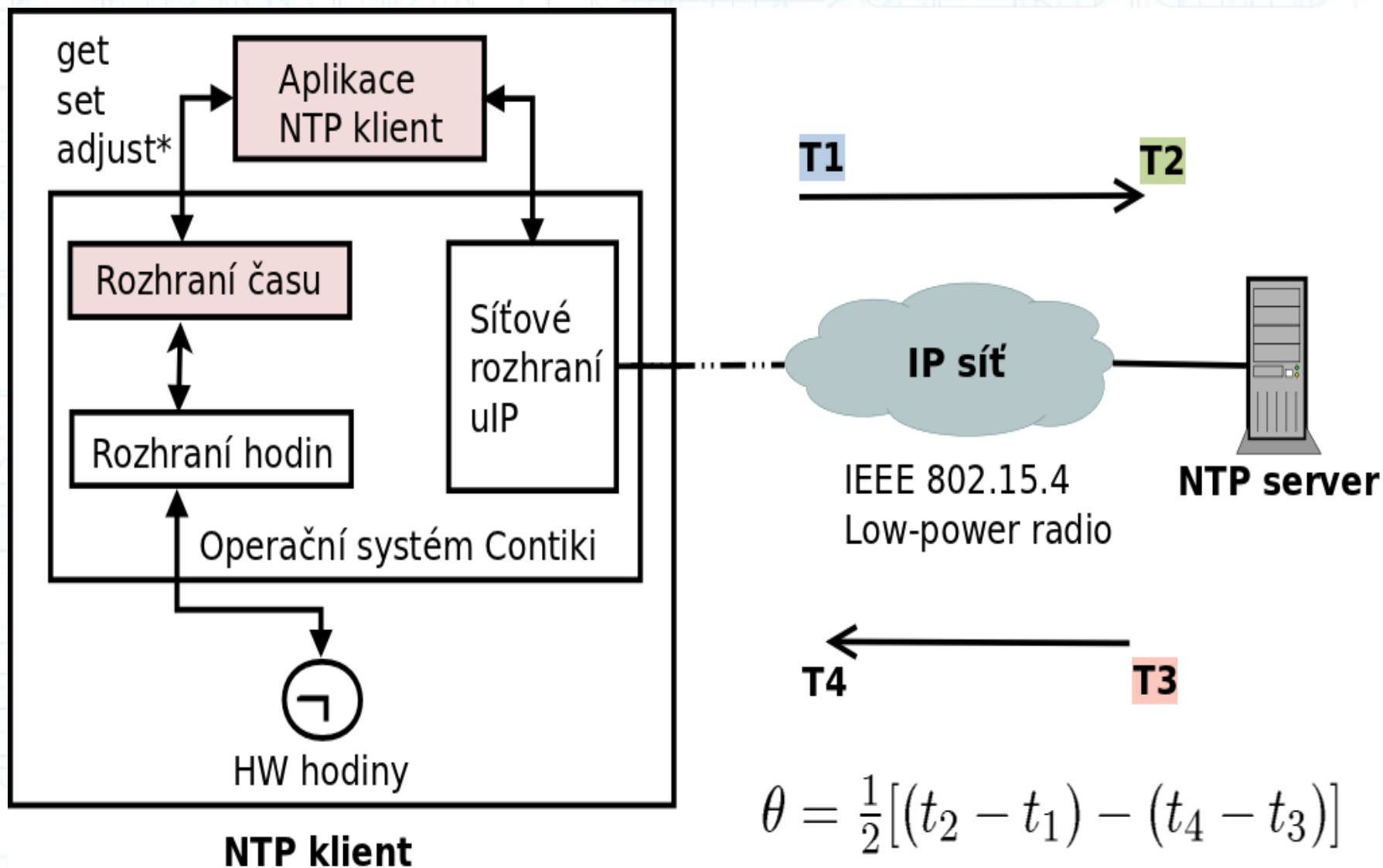


# Schéma klienta



# Rozhraní času

- nastavení – `set_time(sec)`

boottime = `sec` - `uptime`

- zjištění – `get_time(struct timespec)`

`ts.sec` = boottime + `uptime`

`ts.nsec` = `scount` + 

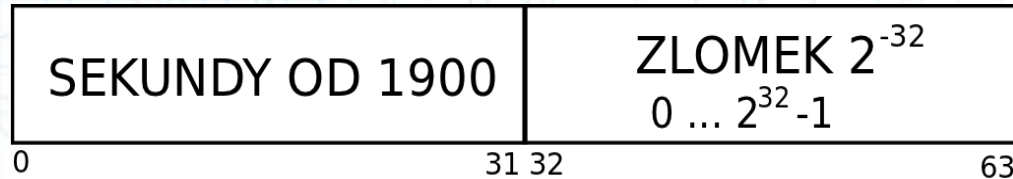
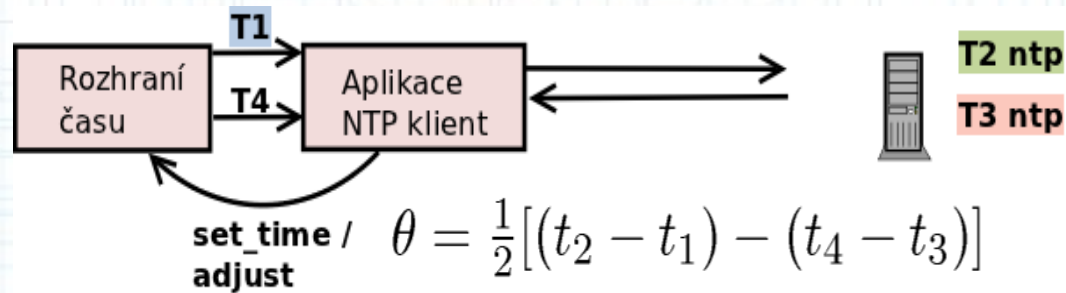
- upravení – `adjust_time(struct timespec)`

adjcompare = vynechané / přidané tiky

STRUCT TIMESPEC	
.SEC sekundy od 1.1.1970	.NSEC nanosekundy

- 8-bit AVR Raven: 244  $\mu$ s - 0,03 s/s
- vliv na Contiki OS

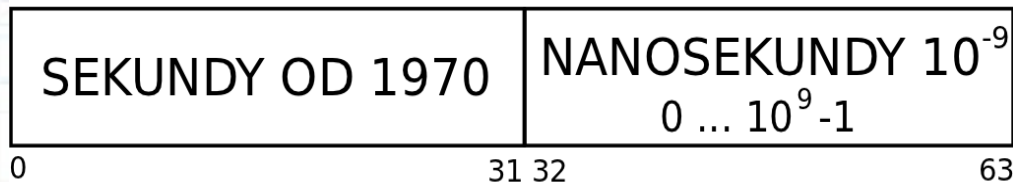
# NTP Klient



$$ts.sec = ntp.sec - JAN_{1970} \quad \downarrow \quad ts.ns = ntp.frac * 10^9 / 2^{32}$$

$$\frac{5^9 * 2^9}{2^{32}} = \frac{5^9}{2^{23}}$$

FP: 3,7KB  
64-bit: 1,0KB



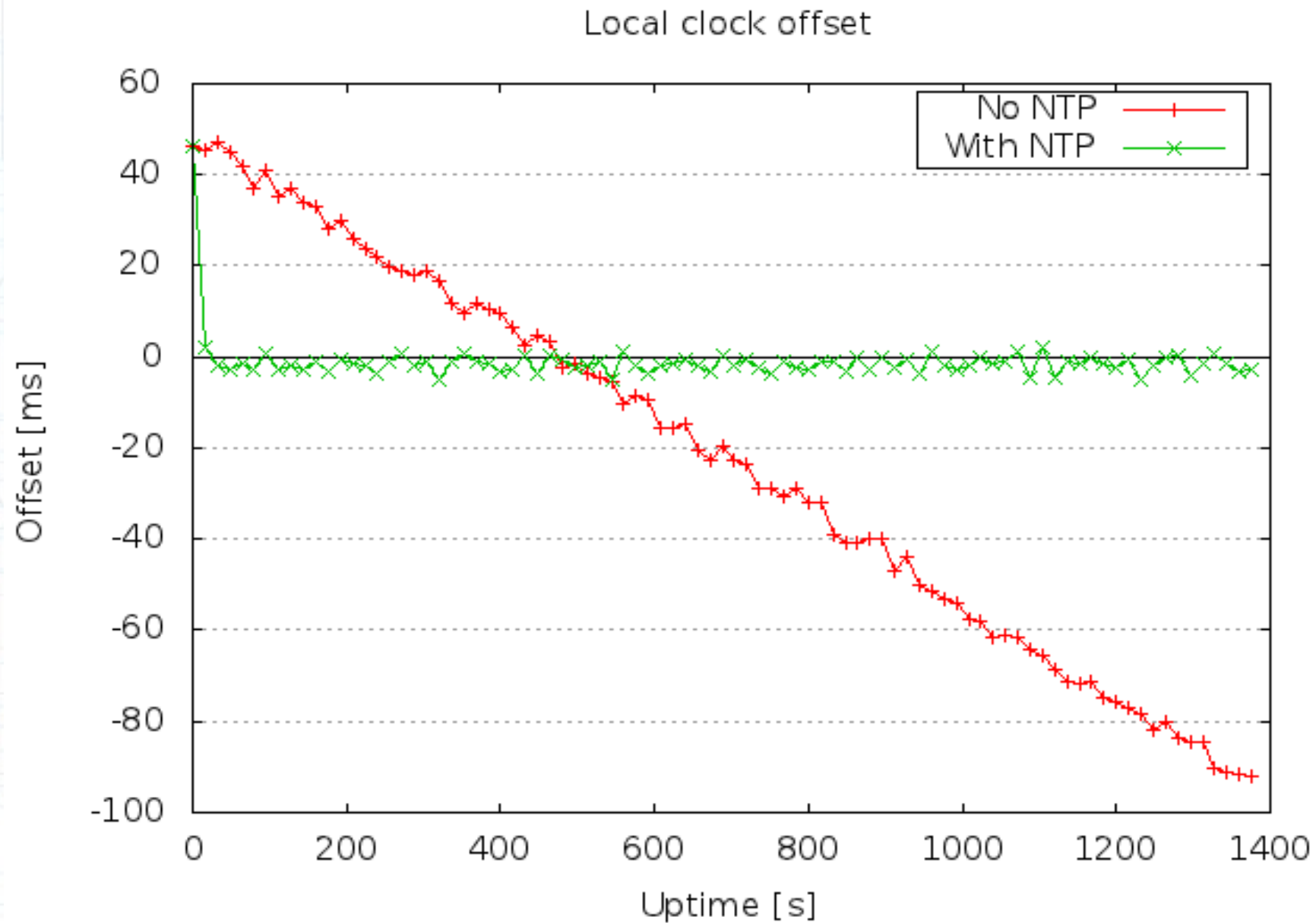
$$(5 * ntp) / 8 = ntp / 2 + ntp / 8 = ntp \gg 1 + ntp \gg 3;$$

$$(25 * ntp) / 8 = 3 * ntp + ntp / 8 = ntp \ll 1 + ntp + ntp \gg 3;$$

→ 300B, 5ns



# Měření



# Závěr

- chybějící rozhraní
- rozhraní času
  - plně přenositelné
  - nejvyšší možná přesnost a rozlišení
- NTP klient – offset < 5ms

# Otázky oponenta – 1/2

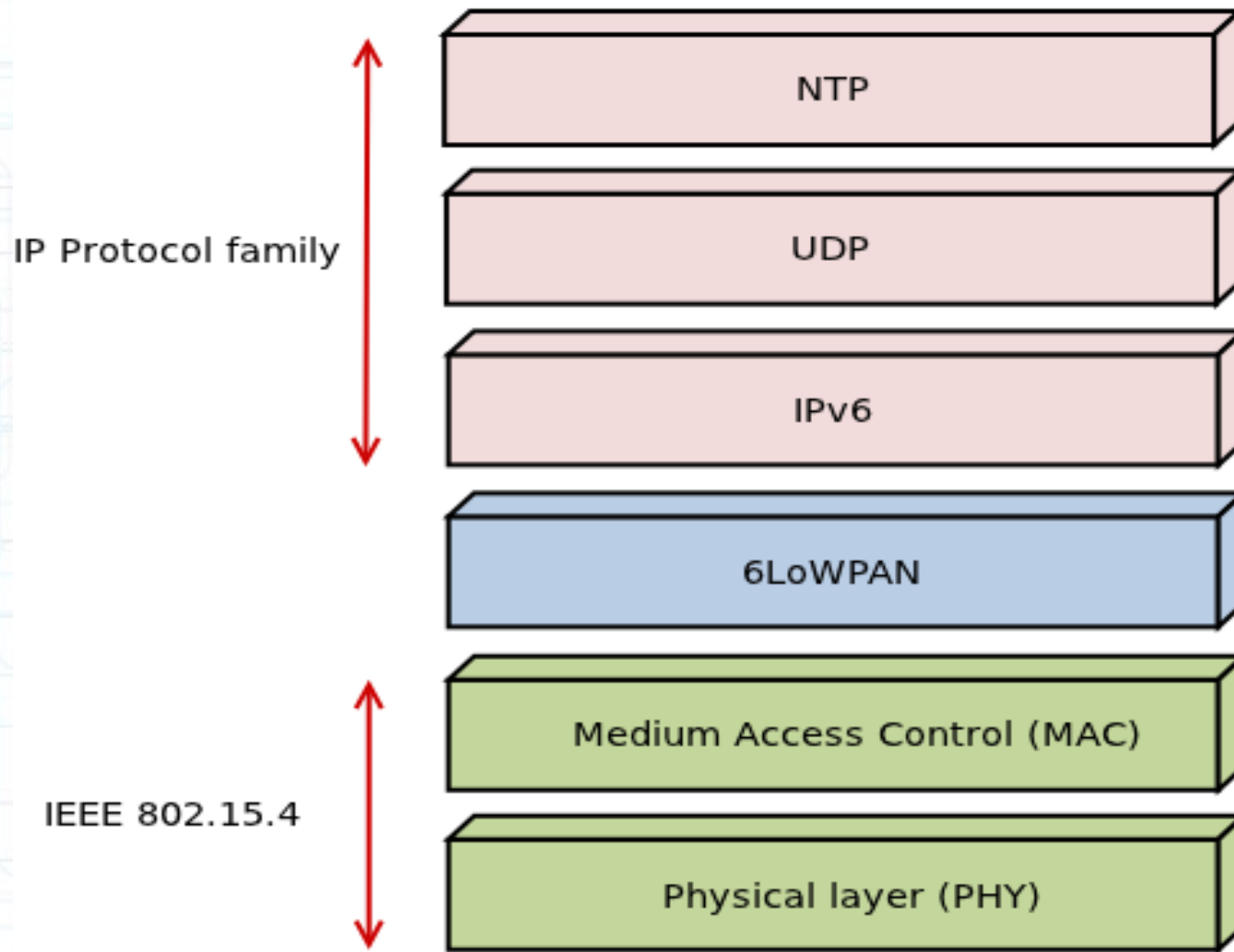
- 1. Jak je v Contiki řešeno ARP a v případě IPv6 NDP?
- ARP v Contiki OS podporuje pouze Ethernet a poskytuje volání pro ovladač síťového zařízení.
  - Výchozí velikost ARP tabulky je 8 záznamů
  - Výchozí životnost záznamu je 20 minut.
- IPv6 NDP v Contiki implementovalo CISCO. Podporovány jsou mechanismy NS/NA a RS/RA (Contiki může fungovat i jako router).
  - Výchozí velikost Neighbor cache jsou 4 záznamy.
  - Pro vypršení záznamů používá časovače typu stimer.



# Otázky oponenta – 2/2

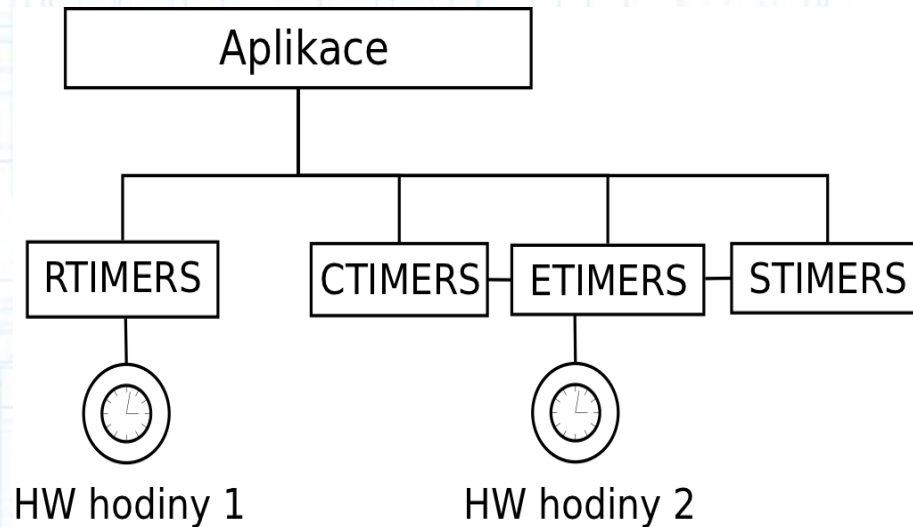
- 2. Zamyslete se nad tím, jak velký vliv na přesnost časové synchronizace může mít protokol ARP/NDP a různé doby vypršení záznamů na straně klienta a na straně serveru?
- Vypršení na straně klienta:
  - dojde pozdržení odeslání NTP paketu na server – 1. slajd
- Vypršení na straně serveru:
  - obdobně

# 6LoWPAN



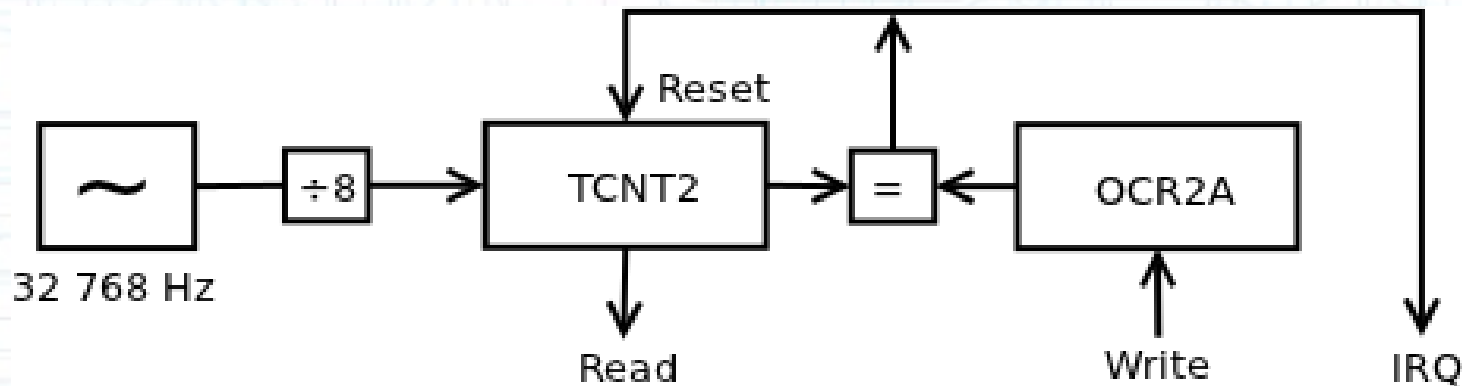


# Časovače v Contiki



- ISR(hw\_clock):  
    count++;  
    if ((++count % CLOCK\_SECOND) == 0) seconds++;  
    ...

# Upravení času



- výpočet množství IRQ s kratším / delším cyklem do adjcompare
- zápis do OCR2A v ISR – zrychlení / zpomalení hodin
- ISR: `scount++;`  
`if ((scount % CLOCK_SECOND) == 0) { seconds++; }`  
...  
`if (adjcompare > 0) { OCR2A_vych + 1; adjcompare--; }`  
`else if (adjcompare < 0) { OCR2A_vych - 1; adjcompare++; }`  
`else { OCR2A_vych; }`