

# Synkronisering av kontroller ved hjelp av GPS

# GPS

- *Sender ut et tidstelegram (Ascii) hvert sekund*  
\$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W\*6A (68 bytes)  
→ Sendes for eksempel på 9600 bits/sekund  
→ 68 bytes x 8 bits/byte = 544 bits →  $T_x = 544 / 9600 = 0,057 \text{ sec} = \underline{57 \text{ ms !!}}$
- *Sender ut PPS (Pulse Per Second) som angir når tidstelegrammet er gyldig (Rising Edge).*

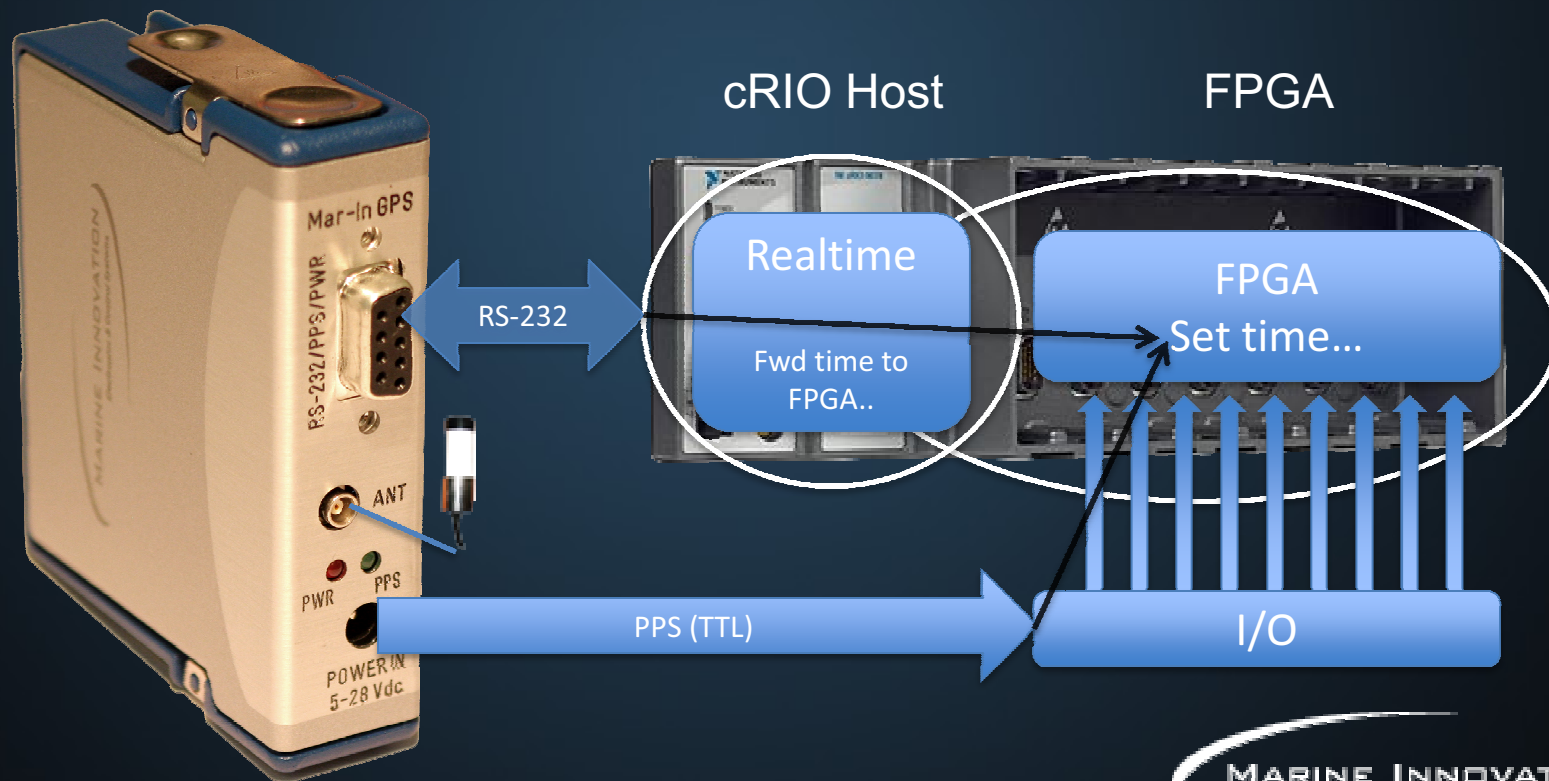


**NB !!**

*PPS mottas før klokkestreng! → Må kompenseres for !  
Skuddsekund – Skuddår – Dato-skifte – Månedsskifte*

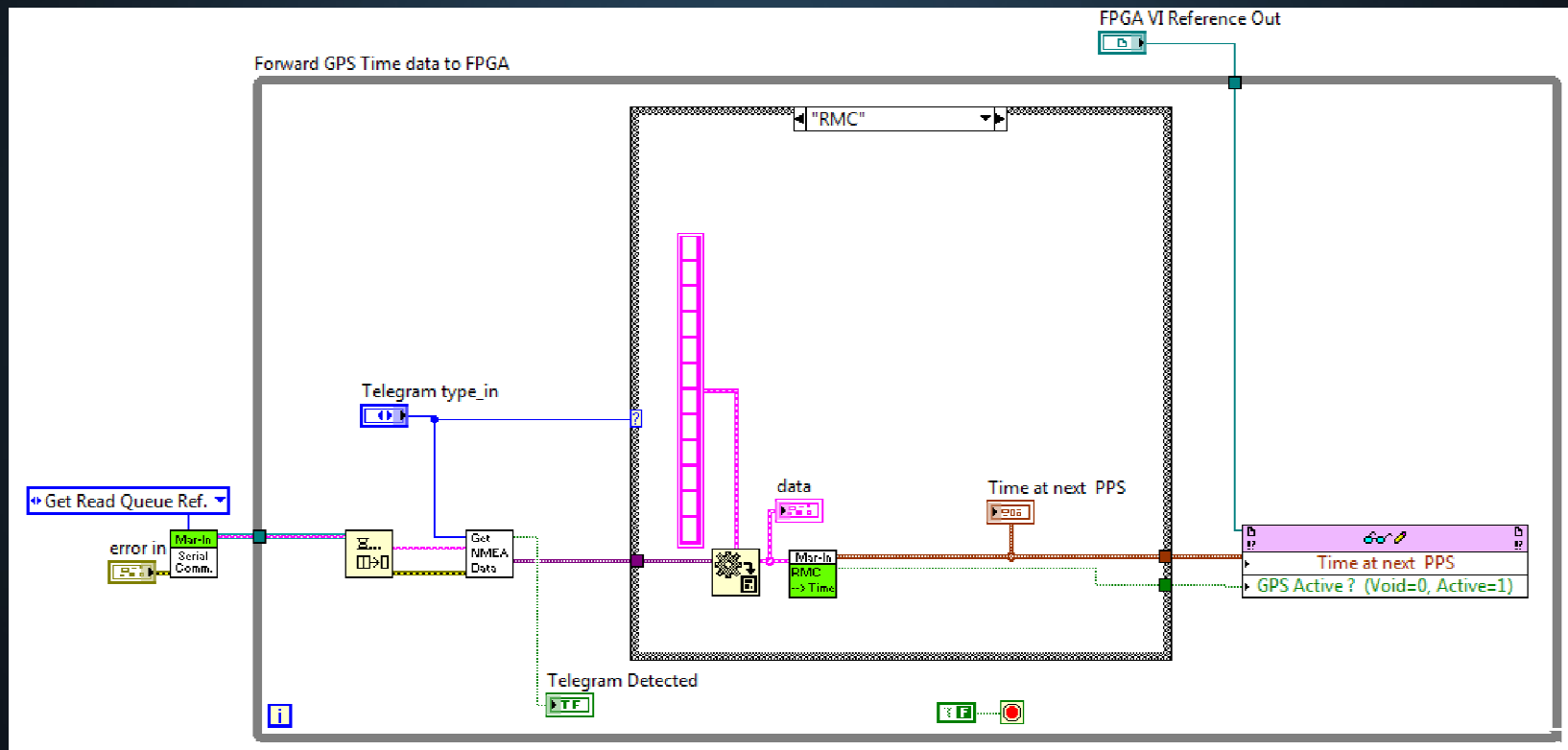
# GPS

- GPS (NMEA time telegram & PPS) ~ +/- 1us absolutt tid
- Digital Input & RS-232 på cRIO
- Korrigjer tiden på hver PPS pulse på FPGA nivå.



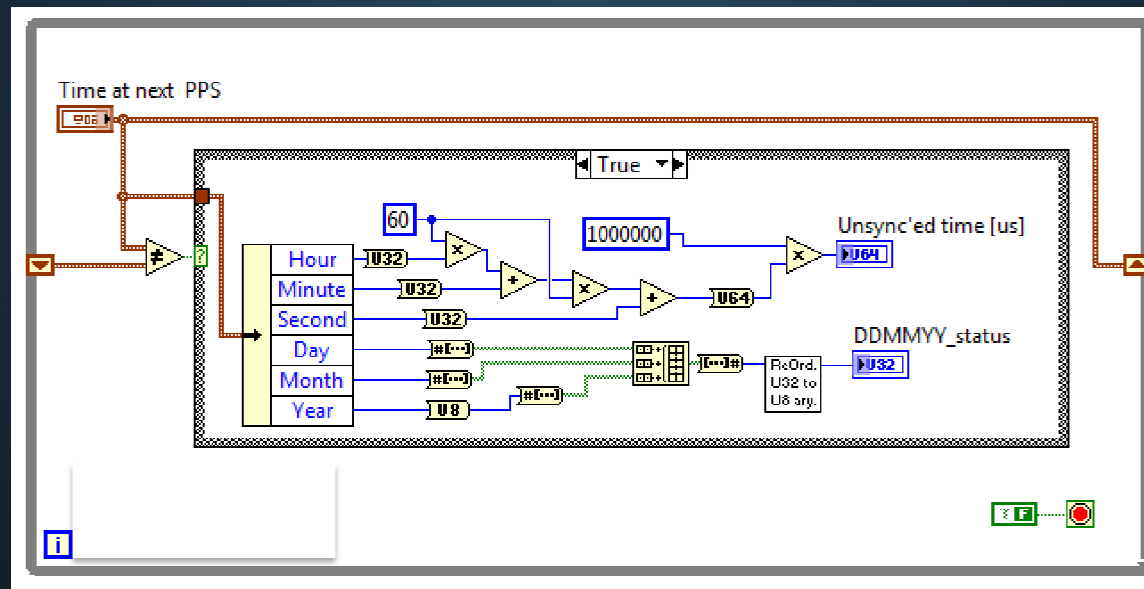
# GPS – Host → Target (FPGA)

- Les serie data fra GPS, og "konverter" til FPGA format...



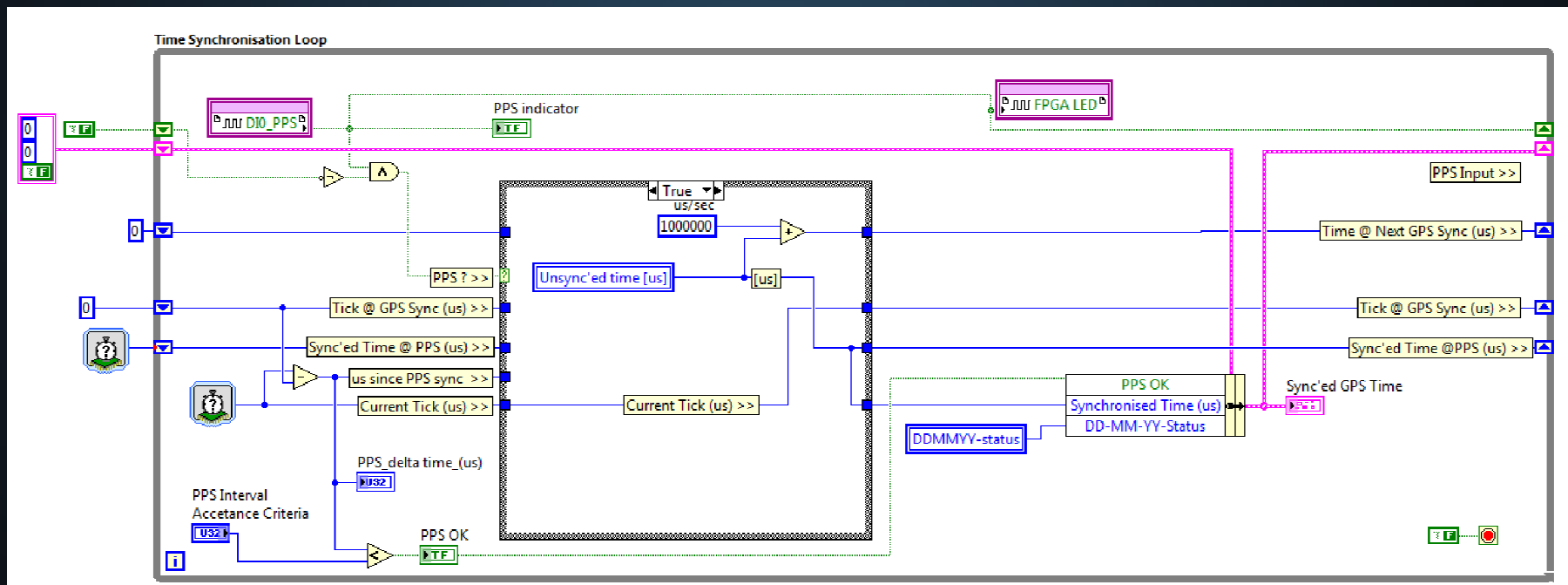
# GPS – FPGA

- Etablerer en "intern" mikrosekund variabel. Denne korrigeres hvert sekund (GPS input), og teller mikrosekunder siden midnatt.
- Den lokale tidsreferansen ligger på FPGA nivå, og sikrer at alle data som blir samlet inn blir tidsstempelt med nøyaktig klokke



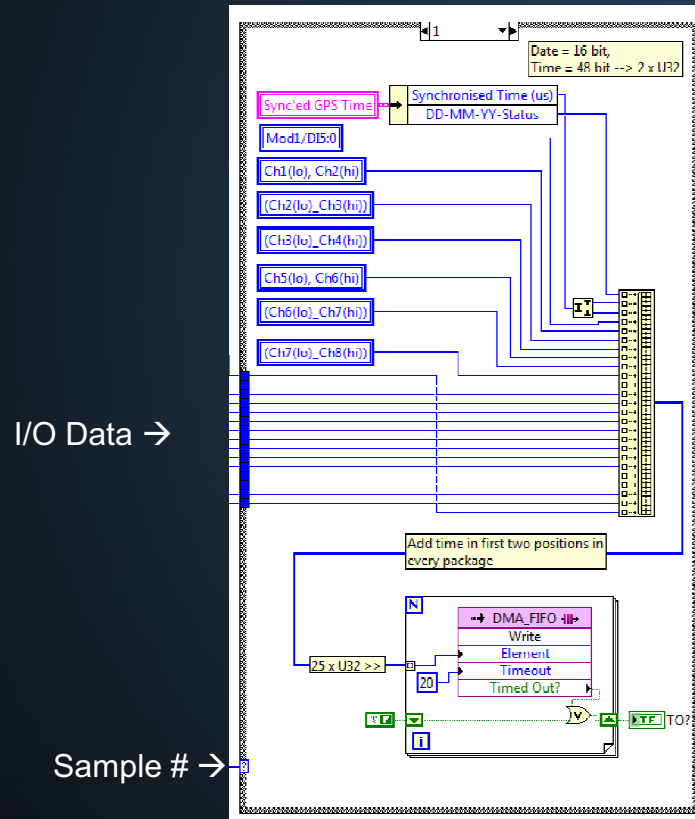
# GPS – FPGA synkronisering

- *Loop spinner med maks hastighet for å detektere "rising edge" på PPS raskest mulig.*

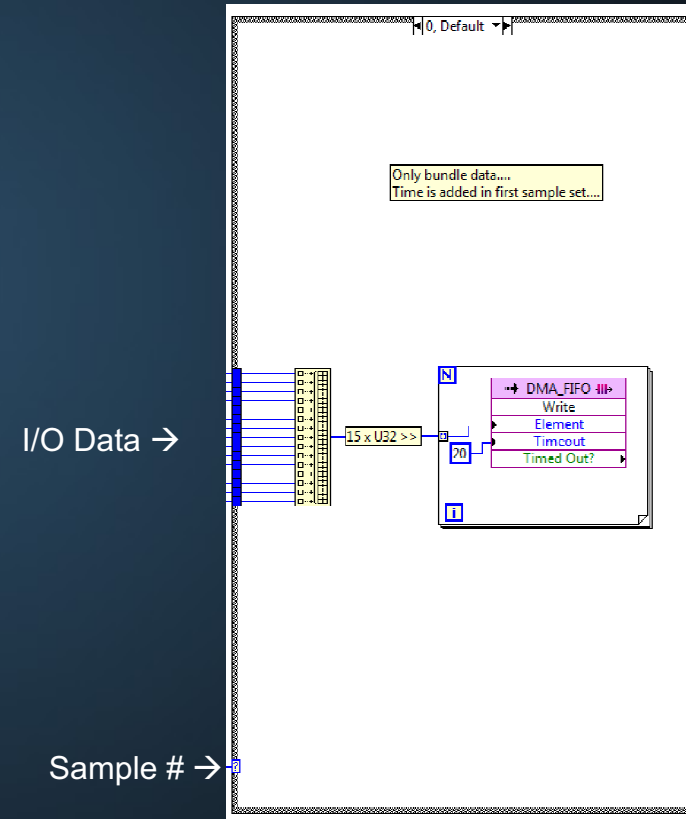


# GPS – FPGA synkronisering

First sample in package



Rest of samples in package



# Datapackage from FPGA - Example

- 20 sensorer, hver 3 bytes @ 10 kHz
- 1 datapakke sendt hvert sekund
- Sample pr. pakke:  $20 \times 10.000 = 200.000$
- Bytes pr. pakke:  $20 \times 10.000 \times 3 = 600 \text{ kByte/s}$
- Hver datapakke har sin egen tidsstempling fra et FPGA-nivå.
- cRIO-host'en legger til TCPIP pakke-ID, og sender dataene til topside-PCen.

Sample No	Sensor 1	Sensor 2	Sensor ...	Sensor 20
"0"	Timestamp @ Sample #1			
1	"Data"	"Data"	"Data"	"Data"
2	"Data"	"Data"	"Data"	"Data"
...	"Data"	"Data"	"Data"	"Data"
10.000	"Data"	"Data"	"Data"	"Data"

- Datapakke:  
"Time stamp + Sample #0 + Sample #X + ... + Sample #10.000"



# Synkronisering av data

## System #1

Sample No	Sensor 1	Sensor 2	Sensor ...	Sensor 20
"0"	Timestamp @ Sample #1			
1	"Data"	"Data"	"Data"	"Data"
2	"Data"	"Data"	"Data"	"Data"
...	"Data"	"Data"	"Data"	"Data"
10.000	"Data"	"Data"	"Data"	"Data"

## System #2

Sample No	Sensor 1	Sensor 2	Sensor ...	Sensor 40
"0"	Timestamp @ Sample #1			
1	"Data"	"Data"	"Data"	"Data"
2	"Data"	"Data"	"Data"	"Data"
...	"Data"	"Data"	"Data"	"Data"
10.000	"Data"	"Data"	"Data"	"Data"

## Synkroniserte data (Topside)

Sample No	Sensor 1	Sensor 2	Sensor ...	Sensor 40
"0"	Timestamp @ Sample #1			
1	"Data"	"Data"	"Data"	"Data"
2	"Data"	"Data"	"Data"	"Data"
...	"Data"	"Data"	"Data"	"Data"
10.000	"Data"	"Data"	"Data"	"Data"

# Synkronisering av kontrollere ved hjelp av Sntp

# SNTP

- Denne funksjonaliteten er innebygget i mange Compact RIO modeller.
- Aktiver funksjonen ved å modifisere ni-rt.ini
- Re-start cRIO
- SNTP funksjonen stoppes hvis man setter klokken manuelt !!!
- **Man trenger en SNTP-server i nettverket for å kunne bruke SNTP klienten i cRIO**

# NTP/SNTP Message format

LI	VN	Mode	Stratum	Poll	Precision
Root Delay (32)					
Root Dispersion (32)					
Reference Identifier (32)					
Reference Timestamp (64)					
Originate Timestamp (64)					
Receive Timestamp (64)					
Transmit Timestamp (64)					
Key Identifier (optional) (32)					
Message Digest (optional) (128)					

*Reference: RFC 4330 SNTP v4*

## Legend

Fields not covered in this presentation

Fields of interest



# Steg for steg eksempel

1. Klienten sender en forespørsel til serveren.
  - Tidsstempelen for sending blir satt til tiden forespørselen forlot klienten.
2. Serveren mottar og svarer på forespørselen.
  - Det opprinnelige sendetidsstempelen blir kopiert til “Originate Timestamp”.
  - Mottak- og sendetidsstempelen blir satt til tiden mottaket blir registrert, og tiden svaret forlater serveren.
3. Klienten mottar svaret fra serveren.
  - Mottagelsestidspunktet blir bestemt.
  - Klienten kalkulerer forsinkelsen og forskyvningen(offset), og oppdaterer klokken sin.

# Tidsforespørseleksempel

Forespørsel fra klient (fields of interest):

Originate Timestamp [0]

Receive Timestamp [0]

Transmit Timestamp [2009-06-11 10:34:22,3]

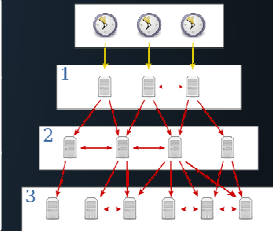


Svar fra server (fields of interest):

Originate Timestamp [2009-06-11 10:34:22,3] (Sendt fra klient)

Receive Timestamp [2009-06-11 10:39:14,5] (Server-tidsstempel)

Transmit Timestamp [2009-06-11 10:39:14,6] (Server-tidsstempel)



Mottakelse av svar fra server:

Destination Timestamp [2009-06-11 10:34:22,8] (Klient-tidsstempel)



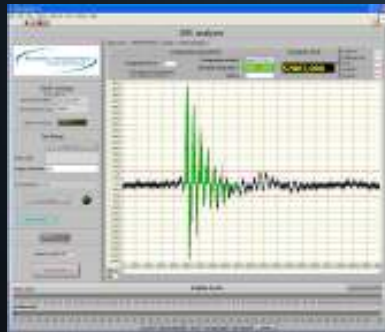
MARINE INNOVATION  
Electronics and Control systems

# Beregning av forsendelsesforsinkelse og systemklokkeforskyvelse(offset).

- Originate Timestamp       $T1 = [2009-06-11\ 10:34:22,3]$       (Forespørsel sendt fra klient)
  - **Receive Timestamp**       **$T2 = [2009-06-11\ 10:39:14,5]$**       **(Forespørsel mottatt av serveren)**
  - **Transmit Timestamp**       **$T3 = [2009-06-11\ 10:39:14,6]$**       **(Korrekt tid sendt fra serveren)**
  - Destination Timestamp       $T4 = [2009-06-11\ 10:34:22,8]$       (Korrekt tid mottatt av klienten)
- 
- Network Roundtrip delay =  $(T4 - T1) - (T3 - T2) = [00:00:00,4]$ 
    - Gjennomsnittlig sendetid =  $0,4 / 2 = 0,2$  sek.
  - System clock offset =  $((T2 - T1) + (T3 - T4)) / 2 = [00:04:52,0]$
  - New Client Time = Current Client Time + Offset  
=  $10:34:22,8 + 00:04:52,0 = \underline{10:39:14,8}$
  - Or:  
=  $T3 + \text{Gj. sendetid}$   
=  $\underline{10:39:14,6} + 0,2 = \underline{10:39:14,8}$



*NB! Formlene er referert til RFC 4330 SNTP v4*



Takk.

[www.kontrollsystemer.no](http://www.kontrollsystemer.no)