

K&K FXE / FXQE80

Company: <http://www.kplusk-messtechnik.de/index.html>



Installation CD "K+K FXE 18-05-08" is in the ResLab (cabinet/computer manual)!

New Counter (FXE07 & FXE04)

Important Notes

1. Es müssen immer **alle** acht Eingänge ein Signal bekommen. Ansonsten gibt es thermische Effekte, die die Zähler die Performance kaputt macht!
 1. Auch wenn man zum Beispiel nur aus 2 Kanälen die Daten speichern möchte und diese im Programm auswählt, werden ständig alle acht Kanäle gemessen!
 2. Auch wenn zum Beispiel die Intervalldauer 100ms beträgt, werden alle 1ms die Daten vermessen und nur alle 100ms gespeichert!
2. Noise floor des Messgeräts ist **5×10^{-13} in 1s** mit Mittelung (= 12ps Zeitauflösung für 1ms Rohmessung)
3. Die acht Eingänge können:
 1. **4kHz - 132MHz** zählen
 2. **~ 0dBm - 17dBm**
4. Sync-Fähigkeit mit PPS
 1. Bei 1V ist Schaltschwelle
5. USB-Nummer: 000000EA (kann verändert werden)
6. Der Zähler kann als TCP-Server über LAN verwendet werden
 1. Nun können bis zu 4 unabhängige User einen Kanal benutzen
 2. Sie können unterschiedliche Modis gleichzeitig verwenden: gemittelt/ungemittelt etc.
 3. Der Server (Netzwerk) wird über die K+K Firmware (befindet sich auf der CD) ausgelesen und der Name/Nickname des Zählers kann angepasst werden
7. Der rote Resetknopf am Frontpanel resetet den Zähler komplett
 1. startet dann mit Default
 2. verwendet man nur, wenn der Computer den Zähler über USB nicht erkennt

Bedienung des Programms



Von links nach rechts



Data Transmission

- **Device**

1. Ein Computer muss mit dem Zähler über USB verbunden sein
2. drei weitere Computer/Nutzer können den Zähler als TCP-Server im Netzwerk auslesen

- **User**

1. Welcher User man ist: 1-4 möglich

- **Chnls** = Channels

1. Anzahl der zu speicherten Channels

- **Intrvl** = Intervalllength

1. Messdauer zwischen zwei gespeicherten Werten

Local TCP Server

- **IP**

1. Auslesen mit K+K Firmware

Log Files

- **Phase**

1. Speicherung der Phasenwerte

- **Frequency**

1. Speicherung der Frequenzwerte

- **Phase Difference**

1. Messung der Phasendifferenz
 1. Ch(n)-Ch(1): Bei allen Kanälen wird die Phasendifferenz zu Kanal 1 aufgenommen
 2. Ch(n)-Ch(n-1): gegen seinen Vorgängerkanal

- **Debug**

Scramble

- **off** = off



- **Auto**

1. geht nur für "Average"-Mode

- **Trimm** = ?

Report Mode

- **Phase Average**

1. Mittelung über Zeit

1. Zähler zählt alle 1ms
2. Beispiel: 1s → Mittelung über 1000 Messungen → Damit wird durch die Mittelung die Messung einen Faktor $\sqrt{N=1000}=30$ besser als nicht "Average"

Freq. Conv. = Frequency Conversion

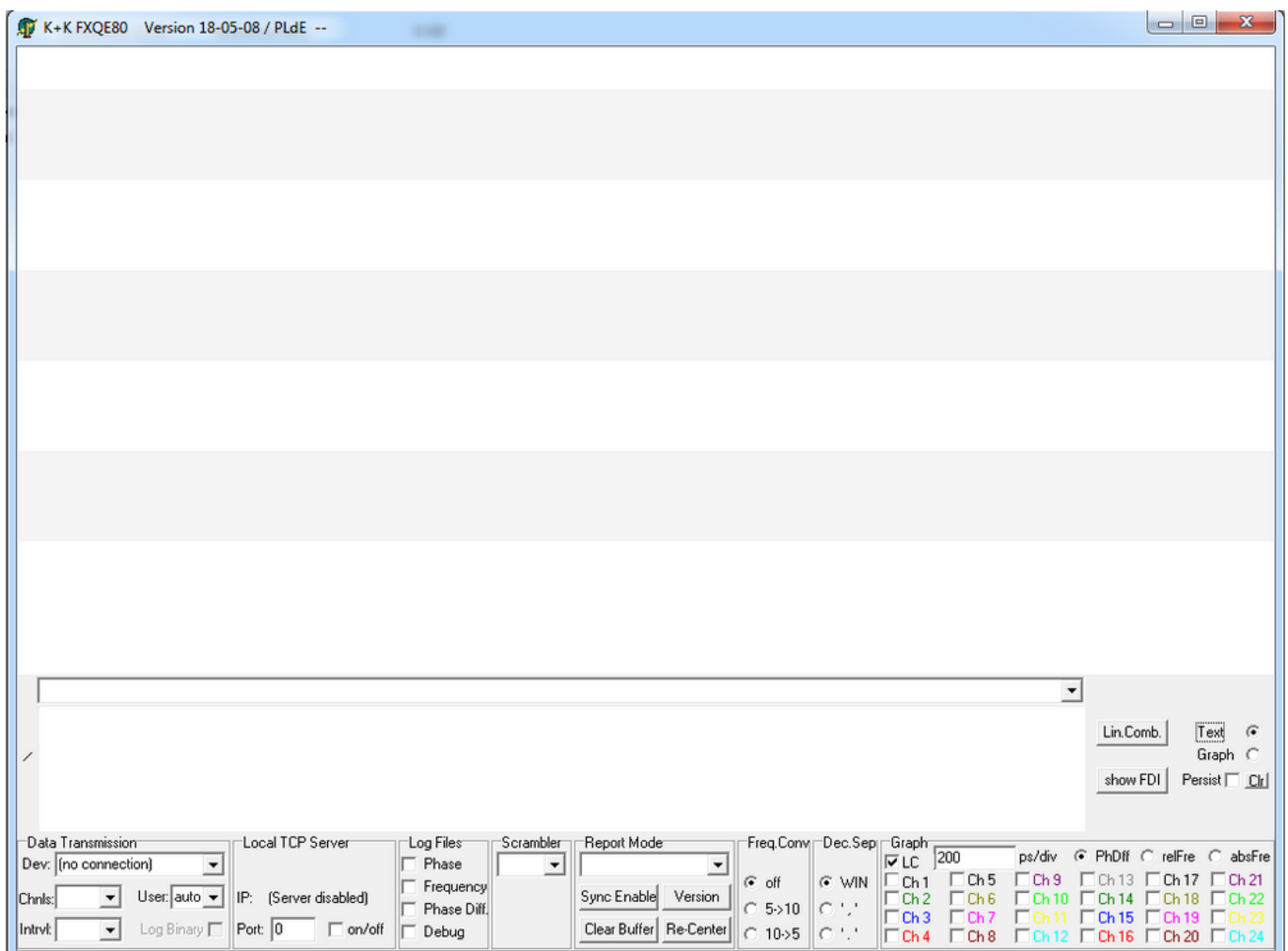
- **off** <fc #ffa500>(immer)</fc>

1. Ist eine Spezialfallanwendung, damit man die Phase zwischen 5MHz und 10MHz direkt vermessen kann

Dec. Conv. = Decimal Separator

Wird zur Separierung der gespeicherten Daten verwendet.

- **W/N** für Windows
- **“,”**
- **“.”** <fc #ffa500>(immer)</fc>



Select your measuring method

Difference between π - and Λ -measurement. Set under *Report Mode*:

- π -measurement = *Phase*
- Λ -measurement = *Phase Average (usually)*

“ Please note that I distinguish between the raw measurement results generated by the hardware, and the reports to the PC which are produced by the microprocessor.

Internally, the FXE counters continuously take raw measurements of the phase (relative to the 10MHz reference) of each channel every millisecond.

The on-board microprocessor can evaluate these raw measurements in different modes for presentation to the PC. I shall try to describe this by giving examples for each mode, where $P_c(i)$ shall be the current raw 'P'hase measurement of channel 'c' at millisecond 'i', and I assume a report interval of 1s (=1000ms):

a) Instantaneous phase report mode: The on-board microprocessor reports to the PC at the selected report interval only the latest raw measurement result. All other intermediate results are ignored. Measurements reported (@ interval 1s) will be

```
Pc( i )
Pc(i+1000)
Pc(i+2000)
Pc(i+3000) ...
```

b) Averaged phase report mode: The on-board microprocessor calculates the average of all raw measurements taken during the report interval and reports to the PC at the end of the interval. All raw measurements contribute to the reported result. This helps to reduce measurement noise. Results reported (@ interval 1s) will be

```
Sum(n=0..999){Pc( i + n )}/1000
Sum(n=0..999){Pc(i+1000+n)}/1000
Sum(n=0..999){Pc(i+2000+n)}/1000
Sum(n=0..999){Pc(i+3000+n)}/1000 ...
```

c) Phase difference report mode: Inter-channel phase differences $P_c - P_1$ are reported. These differences are the 'best possible' measurements, as any uncertainties of the reference cancel and most temperature effects cancel, as they are common to all channels. Measurements reported (@ interval 1s) will be

```
Pc( i ) - P1( i )
Pc(i+1000) - P1(i+1000)
Pc(i+2000) - P1(i+2000)
Pc(i+3000) - P1(i+3000) ...
```

d) Averaged phase difference report mode: Inter-channel differences of averaged phase $P_c - P_1$ are reported. Measurements reported (@ interval 1s) will be

```
Sum(n=0..999){Pc( i + n ) - P1( i + n )}/1000
Sum(n=0..999){Pc(i+1000+n) - P1(i+1000+n)}/1000
```

$$\begin{aligned} & \text{Sum}(n=0..999)\{P_c(i+2000+n) - P_1(i+2000+n)\}/1000 \\ & \text{Sum}(n=0..999)\{P_c(i+3000+n) - P_1(i+3000+n)\}/1000 \dots \end{aligned}$$

e) Frequency report mode: Frequency is Phase advance per time. The on-board microprocessor calculates the difference of the current phase measurement and the measurement taken one report interval earlier, and divides it by the report interval. Measurements reported (@ interval 1s) will be

$$\begin{aligned} & [P_c(i) - P_c(i-1000)]/1000\text{ms} \\ & [P_c(i+1000) - P_c(i)]/1000\text{ms} \\ & [P_c(i+2000) - P_c(i+1000)]/1000\text{ms} \\ & [P_c(i+3000) - P_c(i+2000)]/1000\text{ms} \dots \end{aligned}$$

f) Averaged frequency report mode: Averaged frequency is advance of averaged phase per time. The on-board microprocessor calculates the difference of the currents averaged phase and the averaged phase calculated one report interval earlier. Measurements reported (@ interval 1s) will be

$$\begin{aligned} & [\text{Sum}(n=0..999)\{P_c(i+n)\} - \text{Sum}(n=0..999)\{P_c(i-1000+n)\}]/1000\text{ms} \\ & [\text{Sum}(n=0..999)\{P_c(i+1000+n)\} - \text{Sum}(n=0..999)\{P_c(i+n)\}]/1000\text{ms} \\ & [\text{Sum}(n=0..999)\{P_c(i+2000+n)\} - \text{Sum}(n=0..999)\{P_c(i+1000+n)\}]/1000\text{ms} \\ & [\text{Sum}(n=0..999)\{P_c(i+3000+n)\} - \text{Sum}(n=0..999)\{P_c(i+2000+n)\}]/1000\text{ms} \dots \end{aligned}$$

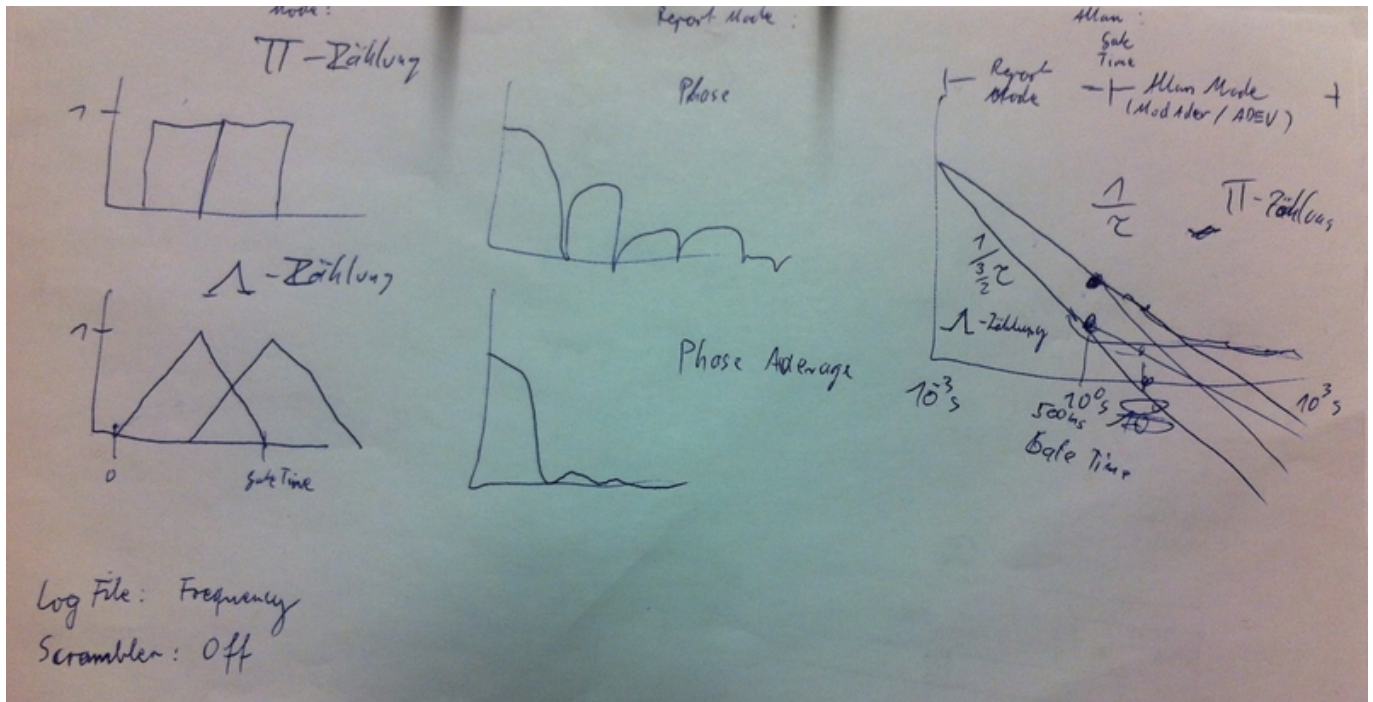
By just re-arranging these sums you can also write

$$\dots = \text{Sum}(n=0..999)\{[P_c(i+n) - P_c(i-1000+n)]/1000\}$$

Hence, averaged frequency can be thought of either as 'advance of averaged phase' or as the average over 1000 non-averaged frequencies (@ interval 1s), in 1ms spacing.

Consequently, data measured during two(!) report intervals contribute to the averaged frequency value reported.

For understanding and communication with PTB



Upgrade der Zählerkarten 2018

1. Die Zählerkarte **FXE02 (alt)** wurde zu einer **FXE07 (neu)** ersetzt. Die FXE02 war zu alt und konnte nicht upgegradet werden.
 1. Sync-Fähigkeit mit PPS
 1. Bei 1V ist Schaltschwelle
 2. Wird als Master-Zähler verwendet
2. Die Zählerkarte **FXE04** wurde upgegradet:
 1. USB-LAN-Buchse
 1. USB-Nummer: 000000EA (kann verändert werden)
 2. Ergänzung der Sync-Verkettung
 3. Wird weiterhin als Slave-Zählerkarte verwendet
3. Allgemein:
 1. Die acht Eingänge können:
 1. 4kHz - 132MHz zählen anstatt bis 65MHz
 2. ~ 0dBm - 17dBm
 2. Der Zähler kann als TCP-Server über LAN verwendet werden
 1. Nun können bis zu 4 unabhängige User einen Kanal benutzen
 1. Sie können unterschiedliche Modis gleichzeitig verwenden: gemittelt/ungemittelt etc
 2. Der Server (Netzwerk) wird über die K+K Firmware (befindet sich auf der CD) ausgelesen und der Name/Nickname des Zählers kann angepasst werden
 3. Verbindung mit dem Computer ist mit USB anstatt Serial möglich
 1. Dadurch können die Kanäle schneller ausgelesen werden. Anstatt 1 Kanal 1ms (bei Serial) bis zu 8 Kanäle jeweils 8 Kanäle 1ms (USB).
 4. Man hat nun die Möglichkeit, dass man einen zeitlichen Verlauf graphisch anzeigen kann

Old Counter (FXE02 & FXE04)

- Possible range: 4kHz - 65MHz
- Recommended Input: 4-17 dBm (1-5V)

Program

Default Settings: 8 Com Ports

Trimmer: Auto Trimmer

Measurement Method: Phase

Intervall length

You can change the length of the *interval*:

- Range: ~1 ms - 1 s
- For shorter length [shortest: 1ms] the memory would fill up very fast.
- Don't forget to adapt the length in the program *Allan-Live with time graph*!

Start a measurement

Push buttons before measurement to erase RAM:

1. *Clear Buffer*-button = Reset
2. Rename the data file in KK_FXE Folder
3. Start new measurement via "Häckchen" at *Frequency*

K+K FX80 / FXE / FXQ / FXC

Header: 1800, Lost: 0

Phase	Frequency	Phase Difference
3374163571617.2659668 Pd	32024005.6568926 Hz	0.5937500 Pd
1996854383475.2776114 Pd	20000000.0514396 Hz	-203602334396.2901924 Pd
5378469280694.6436846 Pd	51013919.6178184 Hz	323028772229.5865436 Pd
3827174769205.7714599 Pd	36292547.3526854 Hz	73405557397.6836711 Pd
2872787900480.1860776 Pd	27315246.7837548 Hz	-79299255342.7034616 Pd

LC: 166645719.4625908 Hz

7020; \$1800
1800; 3374163571617.2659668; 1996854383475.2776114; 5378469280694.6436846; 3827174769205.7714599;
2872787900480.1860776

Text
Graph

Lin.Comb.

Persist

Data Transmission
Serial Port: COM6
Channels: #1...#5
Interval: 500ms

Log Files
☐ Phase
☒ Frequency
☐ Phase Diff.
☐ Debug

Scrambler
ext. uPS:
☐ off
☐ on
☒ Trim

Report Mode
Phase Average
Test Error Version
Clear Buffer Re-Center

Frequency Conversion
(for PhaDiff calculation
from Phase data only)
☒ off
☐ convert 5MHz to 10MHz
☐ convert 10MHz to 5MHz

Decimal Separator
☐ WINDOWS setting
☐ always ','
☒ always '.'

Graph
Scale: 200 E-12/div
☐ PhDiff ☐ relFre ☒ absFre

☒ Channel 1
☐ Channel 2
☐ Channel 3
☐ Channel 4
☐ Channel 5
☐ Channel 6
☐ Channel 7
☐ Channel 8
☐ Channel 9
☐ Channel10
☐ Channel11
☐ Channel12

From:
<https://iqwiki.iqo.uni-hannover.de/> - IQwiki

Permanent link:
https://iqwiki.iqo.uni-hannover.de/doku.php?id=groups:mg:k_k_fxe&rev=1533908647

Last update: 2018/08/10 13:44

https://iqwiki.iqo.uni-hannover.de/

Printed on 2025/07/05 18:07