K+K FXE

High Resolution Multichannel Synchronous Phase Recorder

Manual

Issue: April 2016, Version 16-08-29

K+K Messtechnik GmbH, St.-Wendel-Str. 12, D-38116 Braunschweig

Tel:+49(0)531/501436, e-mail:kplusk@t-online.de

Description of the Multichannel Phase/Frequency Meter "FXE"

The board simultaneously measures the phase of the respective input signals for four independent channels. Here, "phase" refers to the rapidly and continuously growing absolute value, rather than some phase offset relative to a reference signal. Trigger pulses common to all channels are generated at a regular rate of 1kHz by dividing down a 10MHz reference clock input. The measurement results consist of three counter readings for each channel. They are processed by the on-board microprocessor, which performs data reduction by e.g. averaging or computation of phase differences among different channels and reports the results to a PC via the legacy RS232 COM port or USB using a software handshake protocol. A Windows© Dynamic Link Library (DLL) provides for all the protocol handling and data recovery, delivering ASCII coded decimal floating point numbers to the application which are easily readable under all programming languages.

The trigger pulses and the report intervals can be synchronized either to external individual events or to an external clock signal.

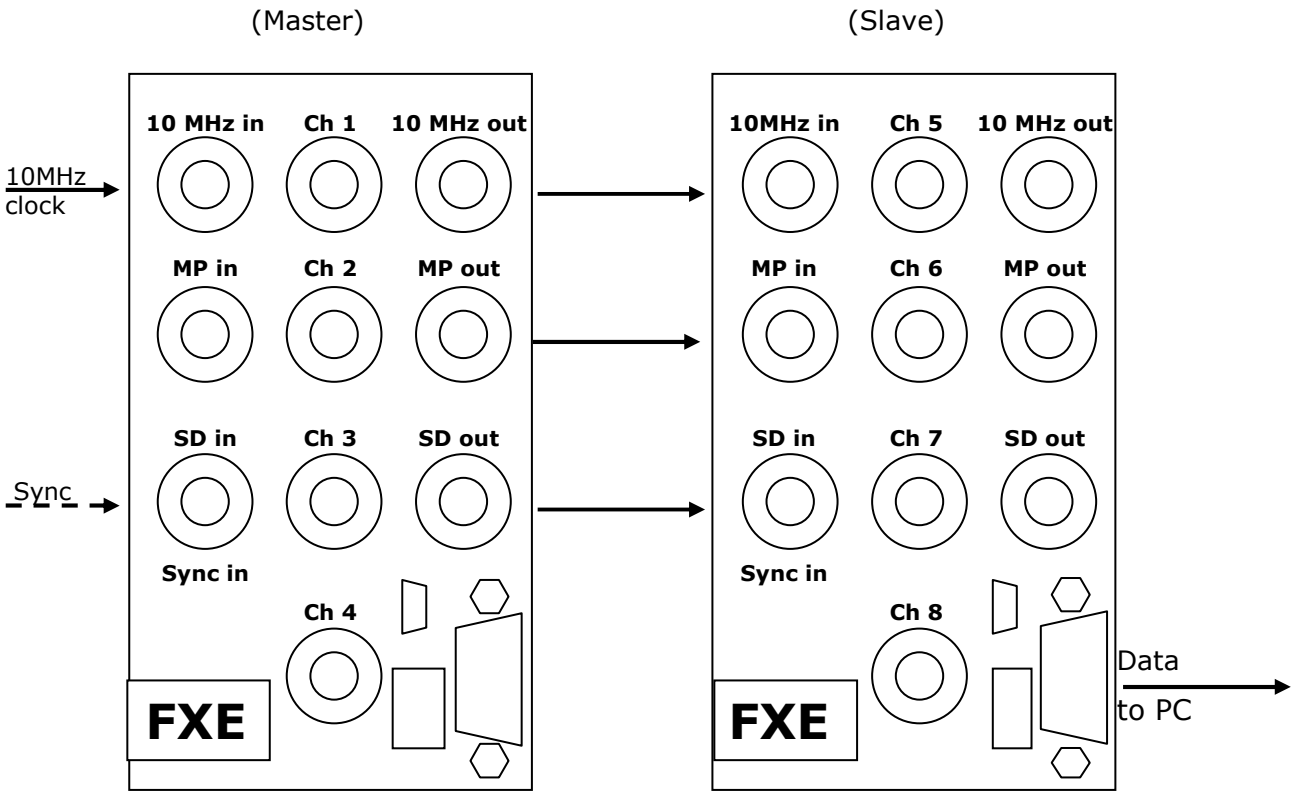
In addition to the counter readings, the data stream reported to the PC also contains control information such as the length of the averaging interval. Thus, the PC application is capable of calculating the input signal's frequency from the advance of the phase values within one report interval. Also, depending on the user's need, frequency ratios between different channels may easily be determined, or some slowly varying phase drift of the output signals of some device under test with respect to the input signal.

Several boards can be concatenated, each of them providing 4 channels, in principle allowing an unlimited number of channels to be measured simultaneously. However, practical limits result from the need to transmit and process all measurement data within one measurement period, thereby limiting the number of channels to 24.

One or more FDI boards may be inserted in a chain of FXE boards, each providing 32 digital inputs and adding as many bytes as a single FXE channel to the data to be transmitted.

For concatenation, the three outputs "10 MHz out", "MP out" and "SD out" of one board are to be connected to the respective inputs "10 MHz in", "MP in" and "SD in" of the successor board. That way, the reference clock, the measurement trigger pulses and the serial data stream are "daisy chained" from one board to the next. The connecting coax cables should be of approximately equal length to preserve the phase relationship among the three signals.

K+K Messtechnik GmbH, St-Wendel-Str. 12, D-38116 Braunschweig, Tel: +49(0)531/501436, kplusk@t-online.de



Only the first board (the "Master", see above) is supplied with the external 10 MHz clock signal to generate (and optionally a sync signal to synchronize) the trigger pulses, while the RS232 or USB data output of the last board only reports to the PC the results from all channels within the chain. Therefore, the last board must have the microprocessor board mounted. The other boards may have a microprocessor mounted, but it will report only those channels from the master up to that respective board.

For highest precision applications, an optional reference phase scrambler SCR may be mounted on the master board, which under control of the microprocessor varies the phase of the reference 10MHz synchronously with the phase averaging intervals, thereby allowing to average over the systematic error residuals of the electronics.

If a scrambler is to be used, it must always be mounted on the master board, irrespective of how many boards may be concatenated. If more than one board is used, three scrambler control signals (Reset, Clock, Up/Down) must be fed via the backplane connectors from the last board carrying the microcontroller to the board carrying the scrambler (please note that these signals are not critical to the measurement accuracy).

Similarly, for the synchronization of trigger pulses and report intervals two sync control signals (Sync, Enable) must be fed via backplane connectors from/to all boards in the chain to ensure simultaneous sync action for all channels.

I. Configuration:

Every board must be configured as either the "Master board" or a "Slave board".
This is done by appropriately setting a jumper close to the upper edge of the board:

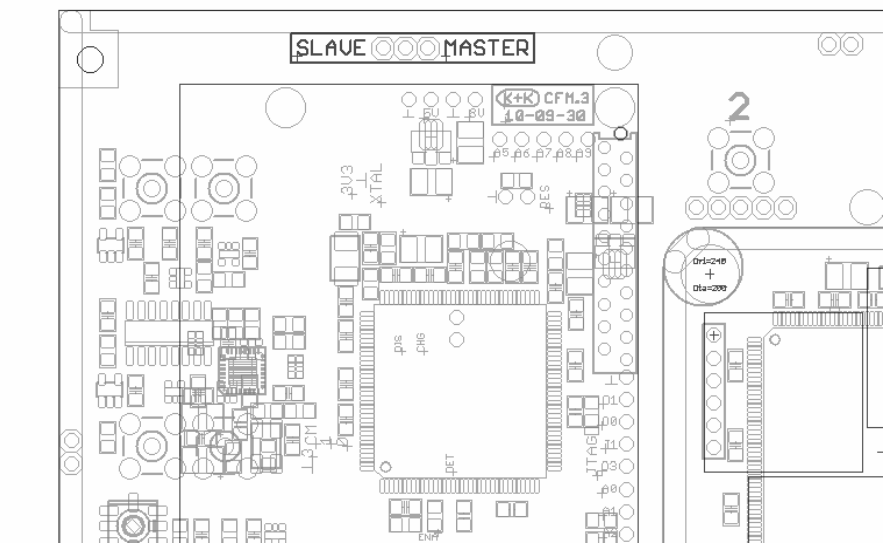


Fig. 2: Position of Master/Slave Jumper

1. MASTER/SLAVE:

On any board used individually, and on the first board of a concatenated chain, the Master/Slave jumper must be set to "Master". This configures the board such that

a) it expects a 10 MHz reference clock signal to generate the trigger pulses which initiate the measurements both in its own 4 channels and in all channels of any succeeding boards;
b) it may optionally scramble the 10 MHz reference phase synchronously with the phase averaging intervals under microprocessor control for highest accuracy.
c) it may optionally accept sync signals.

On all boards which are succeeding the master board in a concatenated chain, the Master/Slave jumper must be set to "Slave". This configures the board(s) such that they expect the 3 input signals "10 MHz in", "MP in" and "SD in" from the preceding board. On slave boards, the 50 Ohm load resistors for the 10MHz and SD inputs must be disconnected.

2. Report Interval, Report Mode, ...:

The report interval, the report mode and other settings are selected by calling the function FX_SendCommand in the KK_FX80E.DLL Dynamic Link Library with the appropriate command byte (see VI. Appendix A below). With the current FXE software version, if averaging is to be used, the averaging interval will always equal the report interval.

II. Data transmission:

Measurement results are reported to the PC by the on-board microprocessor at regular, user settable intervals via a standard RS232 COM port or a USB 2.0 Full Speed connection. In order to exclude transmission errors due to RS232 latency problems within the Windows System, a software handshake is employed which checks every transmitted string for length and check sum and requests re-transmission of any faulty string. In order to allow for maximum report rate, data are transmitted as binary bytes at a rate of 115200 baud. Hence, they are not ASCII readable. In fact, without the proper handshake response of the PC (provided by the DLL, see below), the board will not even transmit measurement data at all.

In addition to checking for errors in the received data, the protocol allows for commands to be sent from the PC to the microprocessor. Such commands are used to e.g. set the report mode (instantaneous phase, phase average, frequency, ...) or the report interval.

All the details of this protocol are taken care of by the DLL, which is delivered with the board. To read reported data or to send a command to the board, the user's application only needs to call the respective DLL routines. Refer to the DLL manual to learn how to call these routines and how to interpret the results returned.

For connection of the phase meter board to the PC's COM port, a standard 'Null-Modem' cable is to be used with female Sub-D connectors at both ends, where the TXD and RXD lines are crossed. Only the TXD, RXD and GND lines are used; the other pins of the board's 9-pin Sub-D socket are not connected.

For a USB connection a cable with a standard 4-pin USB-A connector to the PC and a 5-pin USB mini-B connector to the FXE is needed.

If several boards are concatenated, the output port (USB or Sub-D connector) of the last board (which has to carry the microprocessor board generating the data) will deliver the results from all channels contributing to the data stream.

The serial data stream typically consists of 2 header bytes plus 7 bytes per channel for measurement results. Other messages may have variable length formats. In any case, in addition to these 'data bytes', special 'control bytes' are transmitted encoding the length of the string, its checksum and other protocol control functions. At high report rates, several messages are packed into a single transmitted string to reduce protocol overhead. Such packages are automatically unpacked by the DLL, keeping the entire procedure transparent to the user's application. The data format has been designed to combine the safety of detecting any transmission errors with as little protocol overhead as possible.

III. Optional Reference Clock Scrambler:

Even though great care has been taken when designing the board, there is a small amount of cross talk among the various signals (reference clock, F_x input signals...). This cross talk results in small measurement errors depending on the relative phase of the respective signals, which for specific phase constellations may amount to up to 30ps, thereby limiting the accuracy of the phase meter.

Another limiting effect is the temperature coefficient of the analog pulse length multipliers, which generates errors of similar size.

To overcome these limits, an optional numerically controlled reference clock phase scrambler is available, which is to be mounted to the solder side of the master board. It systematically varies the reference input delay under control of the microprocessor (see VI. Appendix A: list of commands), effectively scrambling the phase of the 1kHz trigger events with respect to the F_x input signals.

The step size of approximately 400ps (100ns/256) is considerably smaller than the width of the structures generated by cross talk. Therefore, the average over all phase measurements taken during a complete scrambler sweep are stable values which no longer depend on the phase of the input signal(s) with respect to the reference clock.

For input frequencies which are not a simple rational multiple or fraction of 10MHz, the scrambler is not needed, as the relative phase(s) will change rapidly by themselves simply due to the difference between the 10MHz reference clock and the input frequencies.

Note that if the scrambler is activated by the appropriate command to the microprocessor ($5x, see below), averaging will be synchronous with a complete scrambler cycle only if the report interval is not shorter than some minimum interval, which depends on the scrambler scheme. This minimum interval is indicated with the description of the scrambler commands.

If a shorter report interval is chosen, the reported phases will cycle systematically through a set of values. In that case, averaging in the PC over as many readings as are being reported during the minimum interval mentioned above will again yield stable results.

In Auto Scrambler mode, the microprocessor will automatically adjust the scrambler cycle to the report interval.

If more than one board in a chain is equipped with a microprocessor, only one of them must control the Scrambler, preferrably the one with the shortest report interval. All other microprocessors must have "SCR off" to avoid conflicting scrambler control signals.

IV. Synchronization:

In order to accomodate different synchronization szenarios (like synchronization to a continuous 1PPS timing signal or synchronization to a single trigger event), a sync logic has been implemented to which the following rules apply:

- As long as the Sync_in signal is low, FXE is in the normal running state. Due to a 3.3kOhm pull down resistor, this is the default for an open input.
- The first rising edge (>3.3V for old boards, >1V since version FXE07 12-03-28) triggers the internal generation of a pulse of 1ms length and disables any succeeding rising edges to have an effect.
- The internal pulse is logically OR'ed with the Sync_in signal. As long as this OR is high, all internal processes of FXE will come to an end within one millisecond, counters are cleared and the system is reset, waiting for the first synchronized raw measurement.
- The falling edge of the OR returns the system into the running state and triggers the first synchronous raw measurement, which is the first contribution to the first synchronous report interval.
- Each microprocessor responds to the sync pulse by sending to the PC a special message "Measurement interval (re-)synchronized!" prior to sending the first synchronous report.
- In addition, immediately before sending the first report in each 1s interval following the sync event, a special message $7F15 will be transmitted. In the current FXQE80.exe application, this causes the time stamp to show an asterisk '*' instead of the usual blank as the separator character between the date and the time.
- By processor command ($0F) the disabled Sync input can be re-enabled for a new sync procedure. This command may be given by any of the processors in a chain, but it will always affect all boards in the chain.

Consequently, for a long (>1ms) positive sync pulse, the trailing (falling) edge will immediately (more precisely: within 100ns) trigger the first raw measurement of the first post-sync report interval.

Alternatively, for a short (<1ms) positive sync pulse, the first raw measurement will be triggered exactly 1ms (with an uncertainty of 100ns) after the leading (rising) edge of the sync pulse.

The following example is an excerpt from the phase log generated by the combination of a (master) FXE with a (slave) FDI board at 1ms report interval, synchronized by a short pulse which also was fed to Bit0 of the FDI board (only a single FXE channel shown for clarity):

```
120828 223525,984 FFFFFFFE        6709267581,2629395

  Measurement interval (re-)synchronized!

120828*223526,000 FFFFFFFF        7518321101,2628174
120828 223526,000 FFFFFFFE        7518331101,2629395

120828 223526,000 FFFFFFFE        7518341101,2629392
120828 223526,000 FFFFFFFE        7518351101,2628174
```

The first report follwing the sync pulse (and every second afterwards) is marked with an asterisk, and the first raw measurement contributing to that report interval is the one which has read the FDI Bit0 as 'high'.

For the sync option to function properly, the pins 30a (Sync) and 30c (Enable), respectively, of all boards must be connected: The Sync signal is driven by the master board only, and must be routed to all microprocessors to synchronize their report intervals.The Enable signal is driven by Open Collector outputs, so any microprocessor - not necessarily the one on the last board - may be used to enable a (re-)sync by issuing the respective command.

Only the master board accepts the external sync signal on its 'SD in' coax connector, while the same connector is a serial data input if the board is jumpered to be a slave.

V. Short description of Inputs/Outputs:

1. All clock or frequency inputs which are to be supplied with user-generated signals, are equipped with an input circuit similar to the one given in Fig. 2 . This applies to the 10 MHz clock input as well as to the channel 1...4 F_x inputs. We recommend square wave signals with 1...5V level as input signals. .

2. As a standard, the inputs have high impedance. If the generator of some signal requires a load impedance of 50 Ohms, an extra resistor has to be added for the respective inputs either externally or on-board. (If it shall be added permanently, a good choice is to solder a size 0805 chip resistor between the center pin and one of the ground pins of the respective SMB connector.)
   Beginning with board version FXE07 12-03-28, on board jumpers can connect the inputs to a 50 Ohm load resistor.

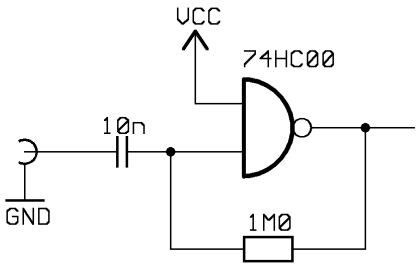


Fig. 2: Input circuit for 10 MHz clock and channels 1...4

3. Every board contains inverting drivers and corresponding inverting input buffers for the transmission of the three "chaining signals" for the concatenation of several boards: 10 MHz reference, measurement trigger pulse (MP), and serial data (SD). Consequently, these three signals appear to be inverted on the connecting coax cables.

4. On old boards with the sync option, the serial data input of the master board is used as a sync input, expecting a nominally 5V logic signal.

   Beginning with board version FXE07 12-03-28, there is a dedicated sync input circuit using a fast comparator, reading voltages above 1V as a logic high level. This circuit provides a jumper selectable 50 Ohm load which must be disconnected for slave boards.

5. The serial output data are provided by a 9 pin male Sub-D connector for transmission to a PC. Only pins 2 (RxD), 3 (TxD) and 5 (GND) are connected. The TxD line is driven by an inverting buffer IC (MAX232) to provide RS232 compatible signal levels (±12V).

6. For the USB port, a 5-pin USB mini-AB socket is mounted to the FXE front panel. With a mini-B cable plugged in, the FXE will function as a measurement device. Future upgrades may later enable the FXE to alternatively behave as a host to e.g. a memory stick.

7. Pinout of the 96-pin backplane connector:

Power Supply:

1 a+b+c: +14...16V (unregulated) in
5 a+b+c: +5V (regulated) out
7 a+b+c: +7V...16V (unregulated) in

Explanation:
The board internally needs both +5V and +12V supply voltages, with most of the current demand of about 400mA (+200mA for the microprocessor, if mounted) arising from the 5V circuitry. Thus it seems reasonable to provide about +8V to the on board 5V regulator (pins 7 a,c). However, if only a single voltage supply is available, it is possible to deliver 14V only to both unregulated inputs. In that case, the heat sink gets pretty warm.

Signals needed only for **concatenation of several boards** via backplane connector:

9a:	SCR_Reset	
9b:	SCR_Busy	(it is recommended
9c:	SCR_Clock	to add a GND connection)
10c:	SCR_Up/Dwn	
12a:	10MHz reference in	normally via Front Panel SMA
12c:	10MHz reference out	contact K+K for solder joints...
14a:	MP in	
14c:	MP out	
28a:	SD in	
28c:	SD out	
30a:	Sync	(it is recommended
30c:	Enable	to add a GND connection)

RS232 port:

26a:	RXD	RxD from PC
26c:	TXD	TxD to PC

USB: normally available at the front panel. Alternatively by special ribbon cable:

b16: DP
b18: DM

2, 4, 6, 8, 11, 13, 15, 17, 19, 21, 23, 25, 27, 32 (each a+b+c) , 29ac: GND

All other pins are either unconnected or reserved - leave them unconnected!

Jumper settings, piggy-back boards:

a) All boards:

Master/Slave: "Master"
"Slave"

b) Master board only:

If no reference phase scrambler SCR is used, the jumper on the solder side of the board must be closed ("DIR") to connect the open scrambler output pin to the scrambler input.

If a reference phase scrambler SCR is mounted, it must be mounted on the master board and the jumper must be open ('SCR)'. If several boards are concatenated, the microprocessor signals SCR_clock, SCR_updwn, SCR_busy and SCR_reset from the last board must be fed via the backplane connectors to the SCR_clock, SCR_updwn, SCR_busy and SCR_reset pins of the master card. In fact, it is ok to connect the SCR_clock, SCR_updwn, SCR_busy and SCR_reset pins of all cards, respectively, as long as only one board (the last one, carrying the microprocessor) drives theses lines and only one board (the master board) has a scrambler mounted.

c) Last board (output) only:

The last board in a concatenated chain of boards delivers the serial data output to the PC. This board must carry the microprocessor board.

d) Sync Option:

For the sync option to function properly, the pins 30a and 30c, respectively, of all boards must be connected. The Enable signal is open Collector, so any microprocessor - not necessarily the one on the last board - may be used to enable a (re-)sync by issuing the respective command. The Sync signal is driven by the master board only, and must be routed to all microprocessors. Only the master board accepts the external sync signal on its 'SD in' coax connector.

The sync scheme expects a positive pulse at the sync input (which is the 'SD in' connector of the master board). For pulses shorter than 1ms, the pulse length is increased to exactly 1ms internally. The reason is that it takes one millisecond to safely finish any current measurement cycle. Any further activity is then suspended for the duration of the pulse. The falling (= trailing) edge of the pulse finally triggers the first measurement after the sync event.

After a sync pulse has been processed by the master board, any other sync pulses will be ignored until a microprocessor has enabled a new sync event. This is done by sending the command $0F via the DLL to the microprocessor.

Preliminary specifications:

- resolution: single shot measurement hardware resolution: 12.2ps
  DLL output string numerical resolution: appr. 30 fs
- stability: ± 100ps absolute, ±5ps inter-channel
- noise floor: 200fs @ 5s report interval, measured at F_x=10MHz
- cross talk: <5ps with scrambler active @ 1s interval
- reference clock input: 10 MHz (optionally 5 MHz with modified scrambler)
- input frequency range: 4kHz ... 65 MHz
- recommended input: 1...5V square wave
- temperature range: 0...+50 °C
- supply voltage: +5V reg. (or 8V unreg.), +12V reg. (or 15V unreg.)
- supply current: FXE: 500mA @ 5V, 50mA @ 12V ;
  µP: 250mA @ 5V ;
  SCR: 20mA @ 5V, 20mA @ 12V .

VI. Appendix A: List of Commands:

The current FXE firmware version accepts the following commands via the DLL's *FX_SendCommand* function:

$0x: control commands:

x=0:	do nothing; if MSBit=1 ($80): clear transmit buffer only
1:	send version string
5:	with MSBit=1 only ($85): enter Flash Program Mode
$A:	reset phase difference counters (effective only for modes 4, 5)
$F:	enable sync input

$2x: set report interval to

x=0:	1 ms
1:	2 ms
2:	5 ms
3:	10 ms
4:	20 ms
5:	50 ms
6:	100 ms
7:	200 ms
8:	500 ms
9:	1 s
$A:	2 s
$B:	5 s
$C:	10 s
$D:	20 s
$E, $F:	(reserved)

$3x: set number of channels to be reported:

x=0:	all channels - this is the same as 'Channels #1 .. #24'
1:	channel #1 only
2:	channels #1 and #2
...	
$B:	channels #1 .. #11
$C..$E:	(reserved as protocol control codes - do not send!)
$0F nn:	channels #1 .. #nn (MSBit must be 0 for this two byte command)

If the hardware provides less channels than are selected by command, all existing channels will be reported.

$4x: set mode to:

x=0:	instantaneous phase
1:	averaged phase
2:	instantaneous phase difference to Channel 1
3:	averaged phase difference to Channel 1
4:	frequency from instantaneous phase advance
5:	frequency from averaged phase advance
6:	raw data for channels 1 & 2 (for K+K development purposes only)
7..$F:	(reserved)

$5x: set scrambler:
x=0: turn SCR off
1..$D: (reserved)
$E: Auto-scheme (adapts the scrambling cycle to the report interval)
$F: trim SCR (1 step of 100ns/256 per report interval)

If in addition to the patterns listed above the most significant bit of any command byte is set, the microprocessor's transmit buffer will be cleared as a side effect, avoiding the subsequent reception of 'old' data.

Other one-byte commands are planned to be implemented in the future as well as longer command strings. Therefore, to avoid conflicts with future firmware and/or DLL updates, the user's application should not send any bytes other than the ones defined above.

VII. Appendix B: List of Strings to be received from the DLL:

All strings returned by the DLL to the application as a result of calling *FX_GetReport* begin with a header consisting of 4 hex digits. This header unambiguously determines the format and content of the string.

As explained in the DLL manual, the most significant digit of the header contains an overflow bit and the 3-bit mode field. This 3-bit field is the same as the one to be given in the set mode command (see above).

0: instantaneous phase reading
1: averaged phase reading
2: frequency from instantaneous phase advance
3: frequency from averaged phase advance
4: instantaneous phase difference to channel 1
5: averaged phase difference to channel 1
6: raw data (for development purposes only)
7: miscellaneous messages

For modes 0..6, the second header digit (=nibbele) encodes the report interval. The digit is the same as the one to be given in the set report interval command:

0: 1 ms
1: 2 ms
2: 5 ms
3: 10 ms
4: 20 ms
5: 50 ms
6: 100 ms
7: 200 ms
8: 500 ms
9: 1 s
$A: 2 s
$B: 5 s
$C: 10 s
$D: 20 s
$E, $F: (reserved)

For modes 0..6, digits 3 and 4 of the header currently contain arbitrary data without any reasonable meaning. However, in future phase-meter firmware versions, these digits might be assigned some specific meaning. Right now, the application should therefore simply ignore these elements.

The data format of return strings with modes 0..5 is explained in the DLL manual, because it is the same for all types of K+K phase meters and expected to remain unchanged with future versions.

Mode 6 is implemented to assist in development work only, may be changed without further notice, and is not expected to be of any use for customers.

For mode 7, which represents various types of one-time messages, the 2^{nd}..4^{th} digits of the header indicate the sub-mode, thereby characterizing the message type and format.

In particular, the following header values have been assigned:

$7000: empty string, transmitted approximately every 100ms (if no other message is to be transmitted) to maintain the protocol running and to allow for rapid command transmission even with very long report intervals.

$7001: Version string; currently contains microprocessor firmware version and DLL version. Shall be extended later to include the FPGA version as well.

$7015: preceedes the first message in each 1s interval since the last sync event.

$7Cxx: Reserved for error messages. It is recommended that strings with a matching header pattern be displayed by the application in some kind of alert window.

$7F51: Measurement interval (re-)synchronized

$7FFE: 10 MHz scrambler PLL error

$7FFF: 80 MHz PLL error

Besides the strings mentioned above, no other messages have been assigned in the current FXE firmware. However, the application should ignore all unknown messages in order to avoid any conflicts if such messages were assigned in future firmware versions.

K+K Messtechnik GmbH, St-Wendel-Str. 12, D-38116 Braunschweig, Tel: +49(0)531/501436, kplusk@t-online.de