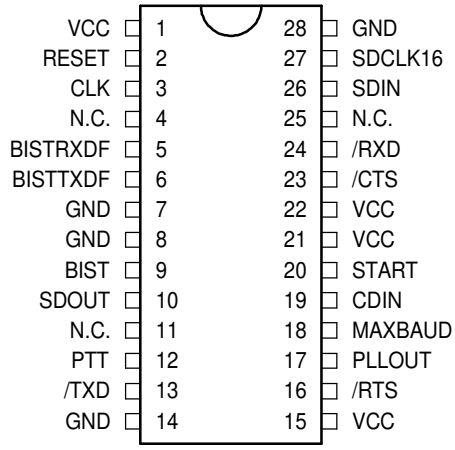
KISSCHIP Euro 180
Synchronous Communications Interface

The KISSCHIP Euro 180 provides a Synchronous Data Link implementing the HDLC-derived Packet Radio Standard AX.25. The KISSCHIP is connected to a Host via a Standard Serial Asynchronous Bus (e.g., RS 232) using the KISS-Protocol for Data-Transmission.
Typical synchronous Transfer-Speed is 300 Baud to 115 KBaud.
Combined with an External Modem and a Transmitter/Receiver the KISSCHIP allows Long Distance Data Transmission in a Network of Multiple Stations.

- Bi-directional Interface between Asynchronous and Synchronous Data Link
- Half Duplex or Full Duplex Operation
- Synchronous Link Implements the Packet Radio Protocol AX.25
  - Sends and Receives NRZI-Coded Serial Data
  - Generates and Checks CRC CCITT-16 Error Detection Word
  - Detects and Generates Break Conditions
  - Integrated 5-Bit Digital PLL for Receiver Clock Regeneration
- Provides Control for Line Access
  - CSMA-Strategy with Programmable Parameters:
    TXdelay, Persistence, Slottime, TXtail
  - Internal Pseudo-Random Generator
  - External Control of Transmission-Start possible
- Asynchronous Link Implements the Amateur Radio Standard KISS-Protocol
  - Allows Software-Control of all Receiver and Transmitter Parameters
  - RTS / CTS -Handshake
  - False Start Bit Detection
  - Digitally Filtered RXD Input
- Integrated Baud Generation
- Large 511 Byte and 63 Byte FIFOs
- Very Low Power Consumption due to 5V 0.7µm CMOS process: 2 mA typ.

DIL Package - Top View

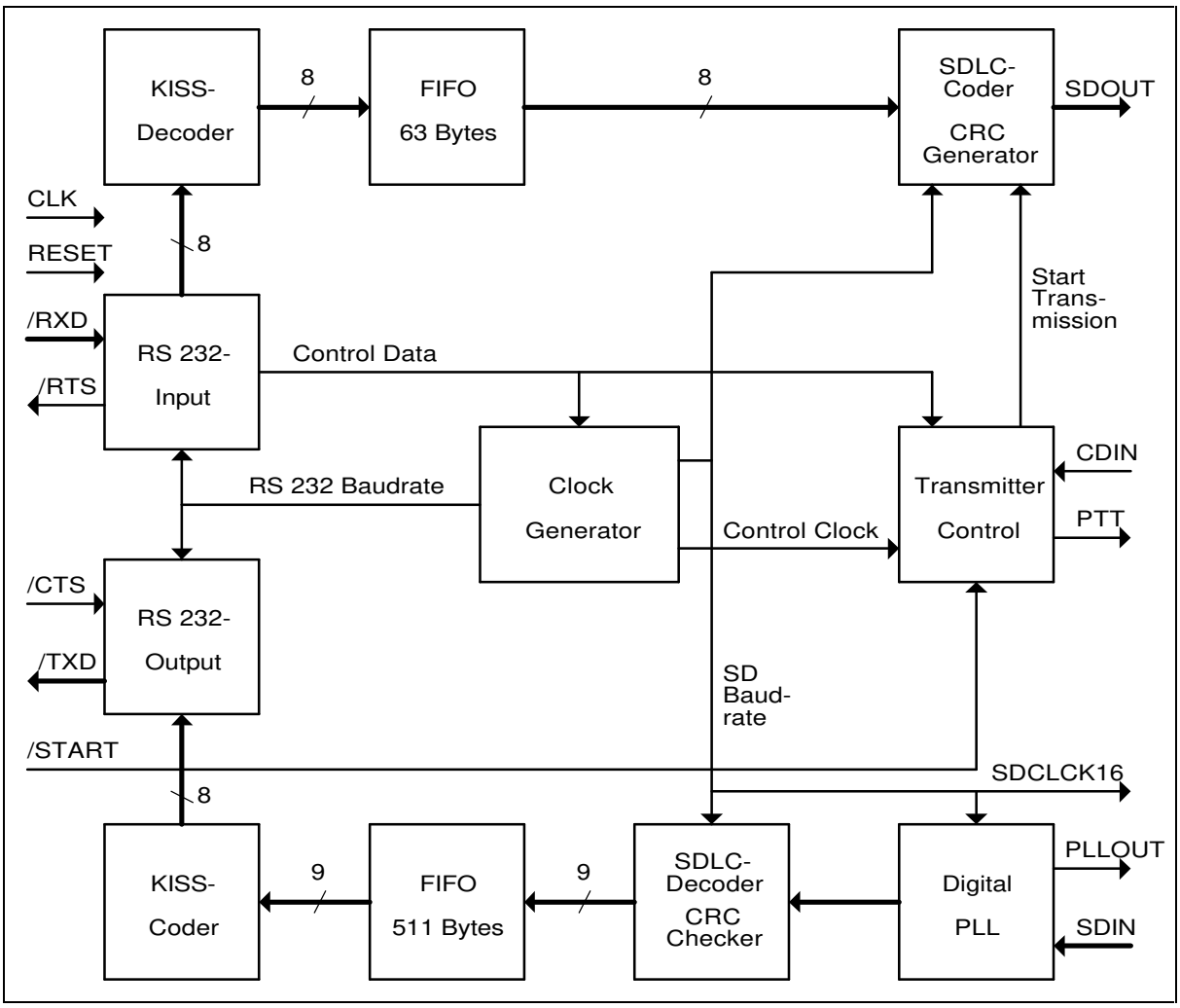


N.C.: No internal connection

Preliminary technical data © B. Dwersteg 1996 - V. 17.01.1996

Universität Hamburg 1995

Technical Data



Block diagram of the KISSCHIP

Preliminary technical data © B. Dwersteg 1996 - V. 17.01.1996

Pin designation and function

PIN NAME	PIN NO	I/O	DESCRIPTION
GND	7,8, 14,28		Common
VCC	1,15, 21,22		+5V Positive Supply
BIST	9	I	FIFO Build-In Self-Test. Tie low for normal operation.
BISTRXDF	5	O	Test output for RXD-input FIFO. Do not connect.
BISTTXDF	6	O	Test output for TXD-input FIFO. Do not connect.
MAXBAUD	18	I	Test input. Sets default values for baud rate generators to 0 on reset. Shortens control clock-divider from 15 bit to 5 bit. Tie low for normal operation.
CLK	3	I	Clock input
RESET	2	I	Positive active reset input
/TXD	13	O	Asynchronous interface output. Set to logic 1 on reset.
/CTS	23	I	Clear To Send. Set to logic 0 to enable TXD data output.
/RXD	24	I	Asynchronous interface input.
/RTS	16	O	Ready To Send. Used to control dataflow into RXD input.
/START	20	I	External control for transmission start if TXdelay-parameter is set to 0. Set to logic 0 to start Transmission.
SDOUT	10	O	Synchronous interface Data output (NRZI-coded)
PTT	12	O	Push To Talk. Use to enable transmitter.
SDIN	26	I	Synchronous interface Data input (NRZI-coded)
CDIN	19	I	Carrier Detect. Set to logic 1 if valid data carrier is detected.
SDCLK16	27	O	16 times SDOUT Clock-Frequency
PLLOUT	17	O	Synchronized SDIN data

All inputs except CLK, BIST and MAXBAUD are sampled with the falling CLK-edge.

Absolute maximum ratings

Supply voltage range Vcc	-0.5 to 7V
Input and Output voltage	-0.5 to Vcc+0.5V
Operating temperature range	-40 to +85 °C
Current into any input	+/-10 mA
Current into any output	+/- 20 mA
Continuous output current	+/- 5 mA

Recommended operating conditions and electrical characteristics

	MIN	NOM	MAX	UNIT
Supply voltage, Vcc	4.5	5.0	5.5	V
Operating temperature	-40		+85	°C
Supply current per MHz fCLK, Io=0 mA		1		mA
Input rise- and fall-time			300	ns
Input voltage low			30	%Vcc
Input voltage high	70			%Vcc
Output high voltage, Io= 2.5 mA	Vcc-0.5			V
Output low voltage, Io= -2.5 mA			0.5	V

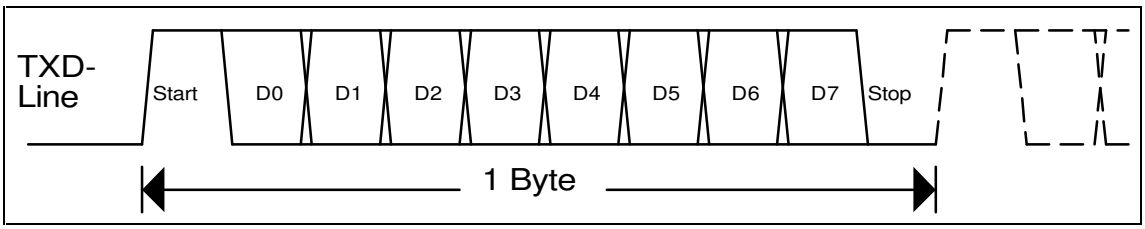
System timing requirements

	MIN	NOM	MAX	UNIT
CLK input low time	30*			ns
CLK input high time	30*			ns
RESET pulse duration	2			CLK cycles

* to be verified

Asynchronous Data Transmission

The asynchronous interface uses standard UART 8 bit, 1 stop bit, no parity protocol. Usage of RTS and CTS pins will be explained in following sections.



Asynchronous data transmission

KISS-Protocol

The KISS-protocol is a very simple standard for data transmission. A KISS block is framed by two FEND special characters. The second byte of each block declares the data type. All following characters up to the closing FEND are user data. FEND characters and FESC characters in the user data are transmitted using a two byte escape code, FESC TFEND and FESC TFESC.

FEND	Data type	data	FEND
1 byte	1 byte	n bytes	1 byte

KISS data frame

Special characters in the KISS-protocol

NAME	USAGE	DECIMAL
FEND	frame end	192
FESC	frame escape	219
TFEND	transposed frame end	220
TFESC	transposed frame escape	221

Data types for the KISSCHIP identified by the character following the starting FEND.

DATA TYPE	FUNCTION	DESCRITPTON
0	User data	Transmission of user data
1	TXdelay	The following byte send to the KISSCHIP contains the TXdelay-parameter. Set to 0 for PTT-control by the START-input. Reset-default: 50.
2	Persistence	The following byte send to the KISSCHIP contains the persistence-parameter. Reset-default: 64.
3	Slottime	The following byte send to the KISSCHIP contains the slot-time. Reset-default: 10.
4	TXtail	The following byte send to the KISSCHIP contains the TXtail-parameter. Reset-default: 4.
16	Asynch.-Baud	The following byte send to the KISSCHIP contains the RS 232 baud rate-divisor decremented by one. Reset-default: 11.
17	Synch.-Baud	The following byte send to the KISSCHIP contains the AX.25 baud rate-divisor decremented by one. Reset-default: 95.

The asynchronous interface baud rate should be chosen as high as possible to minimize delays. The minimum allowable baud rate depends on the synchronous baud rate and the block length. To determine the minimum allowable baud rate consider that FESC-codes in the KISS protocol double the number of bytes which have to be send via the asynchronous interface in the worst case. Further a one byte transmission needs 10 bit times on the asynchronous interface and only 8 or 9 bit times on the synchronous interface. In the worst case the asynchronous baud rate has to be 2.5 times the synchronous baud rate.

Baudrates are generated by dividing 1/32*clock frequency by the baud rate-divisor.
Thus baud rates can be calculated using the following equation:

$$baud = \frac{clock_frequency}{32 \cdot (baud_parameter + 1)}$$

Baud rates for different clock-frequencies

fCLK [MHz]	1.8432	2.4576	3.6864	11.092
default baud async.	4800	6400	9600	28800
default baud sync.	600	800	1200	3600
max. standard baud rate	57600	38400	115200	115200

Preliminary technical data © B. Dwersteg 1996 - V. 17.01.1996

Asynchronous Data Input

The asynchronous data input (RXD) accepts KISS encoded data bytes send with 8 bit, 1 stop bit and no parity.

Wrong escape sequences, multiple FEND or additional bytes after the closing FEND are ignored.

The baud rate of the asynchronous transmitter and receiver is switched immediately after reception of a baud rate control byte. Hence the closing FEND has to be transmitted using the new baud rate.

Only pure decoded data bytes are stored in the RXD-input FIFO. Decoding of input data is done before writing data to the FIFO. The RTS output is cleared when a complete block is inside the FIFO or less than 15 bytes of space are free in the 63-byte RXD-input FIFO. The beginning of the next block can be transmitted to the KISSCHIP even if the previous block is still inside the FIFO and RTS is inactive, provided there is enough space left in the FIFO. But no closing FEND may be send while RTS is still inactive. RTS will return to an active state when all data bytes of the previous block have been transmitted and at least 15 bytes are free in the FIFO. The state of RTS can be ignored for control word transmission.

This behavior can be used for uninterrupted head to tail data transmission:
In the packet radio standard the minimum data block length is 14 bytes.
Transmit up to 15 data bytes to the KISSCHIP at once whenever the RTS-output is active. Transmit a closing FEND only, while RTS is active. Even when RTS goes inactive after transmission of the closing FEND, start transmission of the next data block and send the first up to 15 KISS-encoded data bytes of the block.

Delay times and the operation of the RS 232 output shift register of the external UART have to be considered. So when the transmitter holding register free interrupt of a standard UART is issued, the previous byte and parts of the preprevious byte are still on their way.

The RXD-input FIFO is not allowed to overflow. Otherwise all bytes in the FIFO will be lost.
If more than one block is written into the FIFO, data blocks will be appended.

Asynchronous Data Output

Data is transmitted to the host via the TXD-output using the KISS-protocol. The KISSCHIP will send a 0-type data frame for every valid block completely received from the synchronous input.
The TXD-output FIFO is not allowed to overflow. Hence the CTS-input may not be held inactive for a time longer than the time, needed to fill the 511-byte FIFO completely after reception of a maximum length block, which has not yet been send to the host. The TXD-

output FIFO may hold up to 511 separate blocks. Only pure data bytes are stored in the FIFO. KISS-encoding is done when the FIFO is read.

Synchronous AX.25 Data Transmission

The AX.25 protocol is intended for long distance data transmission between multiple stations with distributed control. Detection of collisions and fail-safe synchronization is guaranteed by the usage of bit-stuffing to distinguish data from the flag-bytes framing each block. The 16 bit CCITT-16 standard CRC is used as checkword.
Implementation of the bit-stuffing and NRZI-coding:
If more than five successive 1-bits are detected in the data stream (or CRC), a 0-bit is inserted. 0-bits are coded as changing level (low-to-high or high-to-low) and 1-bits as constant level.

Flag	Data			CRC	Flag
	Address	Control data	User data		
01111110	112..560 Bit	16 Bit	0..256 Byte	16 Bit	01111110

Packet radio block

Synchronous Transmitter Section

The synchronous transmitter will start transmission only if the PTT output is active (see section on PTT operation). When no transmission is active the transmitter will continuously send flag-bytes.
The KISSCHIP sends data blocks as defined in the packet radio standard. It automatically inserts at least two flag-bytes between each two blocks. The transmitter generates the CCITT-16 CRC check word and appends it to each block send. When the RXD-input FIFO underflows during transmission of a block, a break condition (15 one-bits) will be send and subsequent data bytes will be ignored.

Synchronous Receiver Section

The synchronous receiver will only evaluate blocks when the CD input is continuously active during reception.
On reception, at least one flag-byte is required for separation of each two succeeding blocks. Every received block will be stored in the TXD-output FIFO. Only blocks with correct CRC check word will be send to the host. Due to this block size is limited to the 511 bytes size of the TXD-output FIFO.

Digital PLL

The KISSCHIP contains a 5-bit digital PLL used to recover the synchronous input data clock. The PLL runs with 32 times the synchronous output baud rate and adjusts itself to sample the data input in the middle of the data window. When the synchronous data input SDIN toggles in a bit time, one adjustment of +1/16 or -1/16 bit time is done. Due to this at most 16 input toggles are needed for DPLL adjustment. This corresponds to reception of eight flag-bytes. To guarantee operation the synchronous baud rates of all transmitter- and receiver stations have to be matched tightly. For optimum performance in bad reception-conditions the SDIN signal has to be noise filtered with a bandwidth limited to slightly above the baud rate.

Operation of PTT and synchronous transmission control

The PTT (Push To Talk)-function is intended to enable an external transmitter. The KISSCHIP allows two modes of operation for the activation of the external transmitter and control of transmission start: automatic control by the KISSCHIP or control by the external host.

The KISSCHIP will try to activate PTT whenever a full block is inside the asynchronous RXD-input FIFO or at least 14 bytes of a block have been received from the host. If PTT is already active transmission will start at once.

Automatic PTT activation control

In half-duplex transmission applications the PTT control may be done by the KISSCHIP according to the parameters set by the user:

The CD input will be tested in intervals of 2^15 clock cycles. Whenever CD is detected to be inactive (transmission line is free), a 8-bit random value is read from the internal pseudo-random generator. If the value read is below or equal to the *persistence parameter*, PTT is driven active and transmission is started. Otherwise the KISSCHIP will wait for the time determined by (*slot-time*+1)*2^15 clock cycles and restart the algorithm.

Once PTT is driven active the state of CD will be ignored.

Choosing the *persistence parameter*:

For optimum performance the *persistence parameter* (*P*) should be adjusted dynamically to the number of active stations (*n*).

$$P \approx \frac{n}{2^n} \cdot 255$$

When the number of active stations is very large (>10), a centralized control is advisable.

External PTT activation control

When a bi-directional transmission line is used or a modified protocol is desired, PTT control has to be done externally:

This function is enabled by setting *TXdelay* to zero.

CD will now be ignored for PTT control. PTT will be activated when the START input is sampled active. START is sampled every 2^15 clock cycles.

Transmission of the first data block via SDOUT will start (*TXdelay*+1)*2^15 clock cycles after PTT is driven active.

PTT will be held active as long as data blocks are transmitted. When no more data is present in the RXD-input FIFO PTT will be held active for another *TXtail**2^15 clock cycles. PTT will be disabled afterwards if no further data is to be send. *TXtail* should be chosen long enough to guarantee transmission of CRC and at least on flag-byte.