KEN PROTOCOL - README

By: Ken McCaughey
On: 2025-07-26

Ken's Electronic Network (KEN) Protocol

This serial communications protocol is intended for small micro-controller applications. This data link layer protocol specifies frame elements that define a variety of features. This protocol does not attempt to define all the rules in how they should or could be used. There are considerable permutations and it is up to the user to decide what fits a given application. Not all permutations that can be created are compatible with each other. The intent of this protocol is to have a set of tools and a structure to facilitate communications to meet a particular application. Only implement what is needed for a given application and tailor as needed.

Universal compatibility is not the goal of this protocol. Ease of use and implementation is the goal.

GENERAL FEATURES

- Intended for small micro-controllers.
- Good for point-to-point messaging as well as small networks.
- Borrows ideas from MIDI, BiSync, and S.N.A.P. protocols.
- Features modular/scalable header elements.
  - Trades link overhead efficiency for modularity and parsing simplicity.
  - Three main variations to choose from.
- Designed to be somewhat human readable with a hex viewer.
- Message sizes scalable based on complexity and options used.
  - Can be as short as 3-bytes.
- Supports optional addressing.
  - To, From, and Broadcast
- Supports optional checksums.
  - Several methods supported.
- Supports optional connection and error management.
  - Ack/nack.
  - Connect/disconnect.
  - Supports message sequence numbering.
- Supports frame numbering with total frame count.

NOT INCLUDED

These are the things NOT included or specified in this protocol:

- Physical layer.
- Timing rules.
- Rules for managing connections and error management.
- Header size optimization.
- Byte (or bit) stuffing.
- CRC-32 checksum.
- Mechanism to auto assign addresses.
- Rules to enforce compatibility between different applications.
- Feature for universal plug-and-play like USB or TCP/IP.

VARIATIONS

There are currently three variations. Each has its own specification document. These are not compatible with each other. However they share a lot of commonalities and capabilities that would permit code reuse. Each of these require different techniques to parse the frames. The overhead also varies. Choose the best for your application. See the individual specification documents and the comparison document for more information.

KEN-A, Advanced

This is the fully loaded version. It is primarily oriented towards applications that only need to send ASCII data. There is limited provisions for binary data. This version also permits customization. Header overhead is scalable.

KEN-B, Basic

This is a scaled down version of KEN-A. While retaining the basic features, many of the capabilities are reduced. It still has a scalable header. This variation can send ASCII and binary data.

KEN-C, Compact

This is a compact version of KEN-B. It uses a fixed header of relatively small size. The basic capabilities of KEN-B are retained. This variation can send ASCII and binary data.

KEN-D, Data

This is a minimalist version of KEN-C for just sending data. It uses a fixed header of 2-bytes. Only the frame length and to/from addressing are available. This variation can send ASCII and binary data.