

VisSim/UDP™

Fast UDP Packet Transport Over Ethernet

Key Highlights

- Exchange data between VisSim and any ethernet device
- User selection of up to 256 offsets into a data packet brought out to block pins
- Data types: signed and unsigned 1, 2, 4 bytes integers, 4-and 8-byte floats
- Display data exchange status
- Find transfer function equivalent
- Connect to multiple IP addresses
- Receipt of message
- Transmit timing control

System Requirements

- Professional VisSim v9.0
- Windows XP, Vista, 7, or 8
- Internet connection
- 128 MB RAM
- 125 MB hard disk space

Introduction

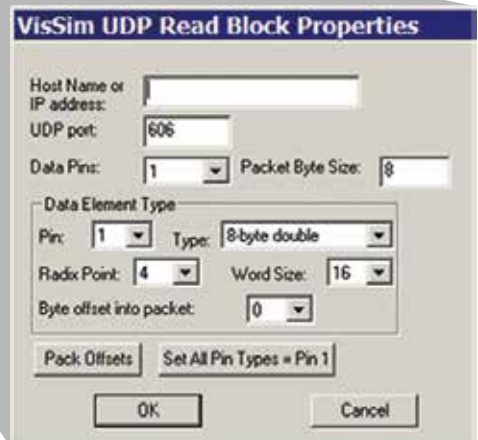
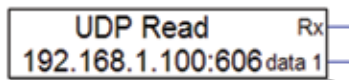
VisSim/UDP lets you read and write standard UDP internet packets from a VisSim block diagram using UDP Read and Write blocks.

With VisSim/UDP, you can determine the packet layout by specifying the byte offset and datatype of each element. You can choose 8, 16 or 32-bit signed and unsigned integers; 4 byte and 8 byte floats; and 16- or 32-bit scaled integers.

You can use VisSim/UDP for remote sensing, data exchange, and remote control. You can also easily log data, plot, strip chart or run a virtual plant or controller for model based development.

To exchange UDP data with a VisSim model, you must have an internet connection, either wired or wireless. Then, with UDP Read and Write blocks wired into your diagram, you can begin exchanging data by starting a simulation.

You can transmit and receive from any number of different IP addresses in a single diagram. The UDP Read and UDP Write blocks let you read data from and write data to UDP ports. Data types and byte offset within the data packet can be quickly modified block dialog boxes.



Properties dialog box for the UDP Read block. You can easily update byte offset and data type for each element.

We selected VisSim/UDP to help us develop inter-device UDP communication for our automated glucose measurement system. The ability to control the packet size and UDP address, and interpret the size and type of data bytes within the packet, gave us the flexibility we needed right out of the box.

Adam Fettig, Software Engineer, Luminous Medical