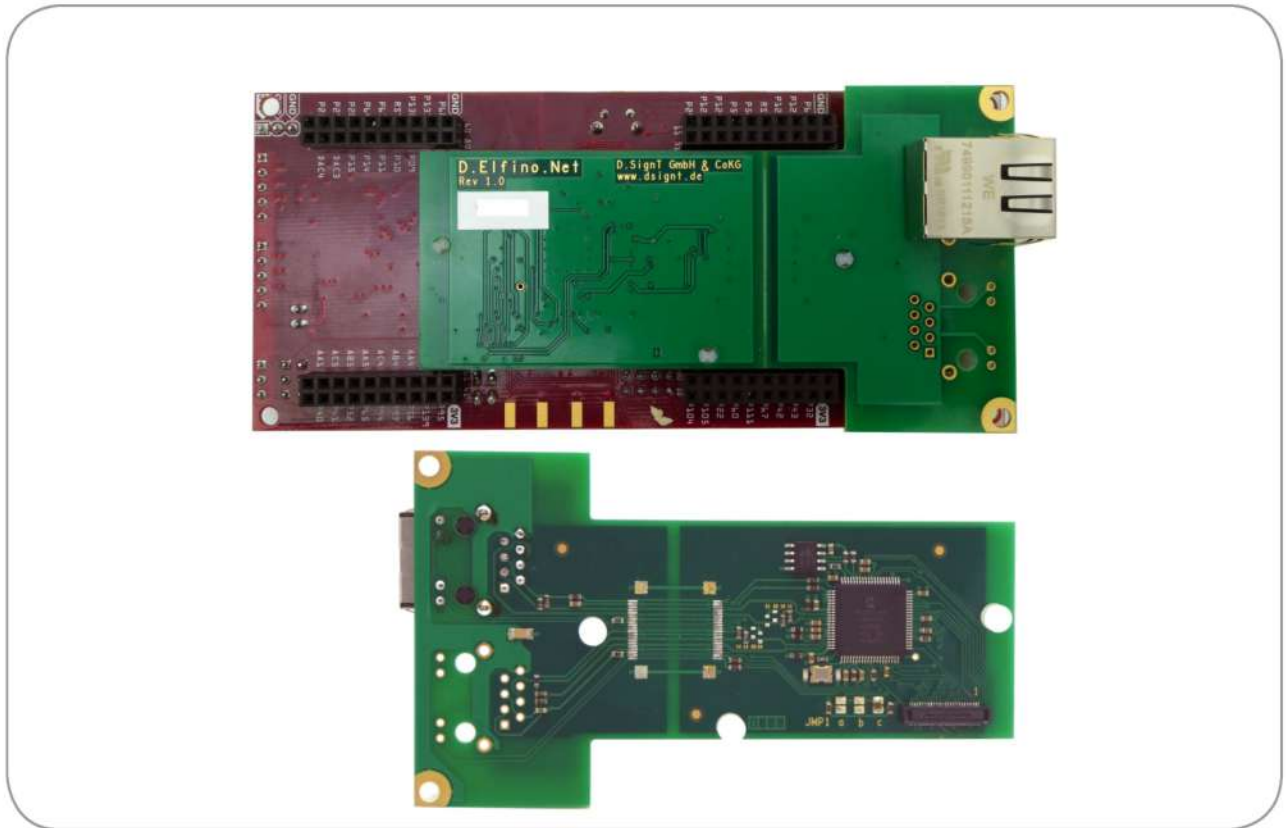


D.Elfino.Net

Board Revision 1.0

Document Revision 1.0

Feb 22, 2019



Key Features

- Ethernet add-on card for Texas Instruments Delfino™ processors
- Fits LaunchPad™ (LAUNCHXL-F28379D) and controlCARD (TMDSNCND28379D)
- 1 or 2-port 100Base-Tx / 10Base-T with auto MDI/MDIX, optional 100Base-FX fiber ports
- No additional power supply required
- 16-bit memory-mapped EMIF interface
- Variable installation options:
 - if required, the RJ45 socket part can be separated and extended with FFC / FPC cable
- Highly optimized TCP/IP stack for C2000™ real-time control MCUs

D.SignT TCP/IP Stack

The TCP/IP object code library has carefully been tailored to meet the constraints of a DSP system. Code and data memory size have been minimized, and no additional resources like DSP interrupts or timers are required. The TCP/IP protocol stack can be used in a 'bare-metal' C program, just as running as a task in the Texas Instruments DSP/BIOS and SYS/BIOS real-time operating systems.

The TCP/IP stack supports the following protocols:

- ARP - Address Resolution Protocol, resolves the IP address to a hardware MAC address. No user-action is required. If an address is unknown, an ARP request is generated automatically.
- IP - Internet Protocol. All data transferred by DNS, DHCP, ICMP, UDP and TCP is automatically packed into IP packets.
- ICMP - Internet Control Message Protocol. The protocol stack responds to "ping" requests to test a connection.
- UDP - User Datagram Protocol. UDP provides a one-to-one or one-to-many connectionless data path. Data transmitted via UDP is not guaranteed to reach it's destination. This protocol has very low overhead and is especially useful for transmitting non-critical data like audio and video data.
- TCP - Transmission Control Protocol, provides reliable, connection-oriented, one-to-one connections. All data is acknowledged by the receiver and re-transmitted automatically if required. This protocol should be used for critical data like software uploads, commands, etc.
- DHCP - Dynamic Host Configuration Protocol. This protocol has been developed to ease maintenance of a TCP/IP network. A DHCP server manages the allocation of IP addresses and provides additional network configuration data like gateways, DNS servers etc. The TCP/IP stack integrates the client functions required to obtain an IP address, DNS server, and gateway.
- DNS - Domain Name System. This protocol allows to use symbolic host names instead of numerical IP addresses. The TCP/IP stack integrates the client functions to query a DNS server to resolve a host name.

Higher level protocols like SMTP, HTTP or FTP are based on the described protocols, most of them using TCP. The TCP/IP library includes a ready-to-use FTP server, which allows to upload programs and parameters to Flash Memory, or download logs and data files. The FTP server is widely configurable: users, passwords, directories, files, and access restrictions are maintained in a simple data structure. A HTTP server framework is also provided. This framework handles multiple connections and passes GET and POST parameters to a user-defined callback function, hence providing the required flexibility for dynamic data. The DSP can send static HTML pages and images as well as inserting the current value of variables, generate images from data acquisition buffers, etc. on demand. SMTP functionality is provided to send an e-mail, e.g. to periodically send log-files to the system administrator. Finally, a Modbus TCP implementation is provided for process control.

The TCP/IP software uses a socket architecture, similar to the familiar Berkeley sockets. Following is a description of the function calls implemented:

General Initialization

net_init() initialize sockets
 net_set_gateway() configure gateway for connections outside the local IP net.

Socket Configuration

socket_open() open a new socket
 socket_close() close a socket
 set_socket_option() specify non-standard socket options, e.g. disable UDP checksum
 socket_define_callback() install a user-defined callback function
 install_icmp_socket() install a socket and buffer for 'pings'

Send Functions

net_send() non-blocking send function
 net_send_string() blocking send function for strings
 net_send_ready() blocking send function for binary

Receive Functions

net_recv() non-blocking receive function
 net_recv_ready() blocking receive function

Miscellaneous Functions

net_isq() main network polling function, must be called periodically in your program's main loop or from a periodic task.

TCP Connection Functions

connect () establish a connections
 shutdown () shutdown a connection
 gethostbyname() host name resolution
 accept() test if TCP socket is connected

Ordering Information

D.Elfino.Net-Card	Board
DS.D.Elfino.Net	Development Package including Board, Support-Software and TCP/IP Stack development license
QL.D.Elfino.Net	small quantity TCP/IP stack license
OL.D.Elfino.Net	unlimited OEM license for the TCP/IP stack



D.SignT GmbH & Co. KG

Marktstraße 10 • 47647 Kerken • Germany
 phone +49 (0)2833 / 570 976 • fax +49 (0)2833 / 3328
 info@dsignt.de