ASIX®

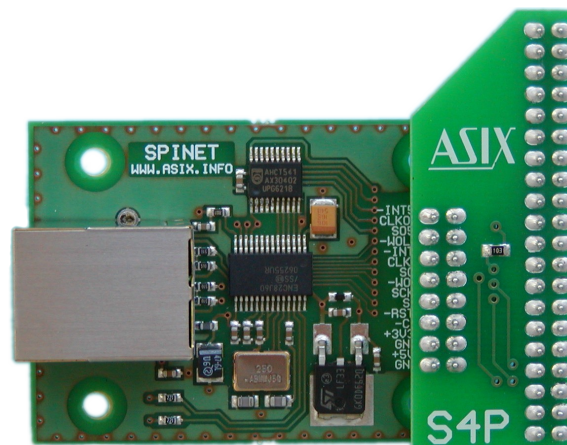# SPINET4PVK

Ethernet evaluation kit

User's Guide

- SPINET Ethernet module
- Adaptor for easy connection to PVK40 board
- Example program in C

TABLE OF CONTENTS

Note: This document contains hyperlinks pointing to web pages on the Internet. If the links happen to be broken (pointing to a non-existing page), please download recent version of this manual from www.asix-tools.com.

1. USING THE KIT AS IS

The provided HEX files are targeted for use with PVK40 board in following configuration:

SPINET	PVK40	J1 pin
+5V	+5V	18,21
GND	GND	20,23
SCK	C3	35
SO5	C4	36
SI	C5	34
-CS	B6	4
INT5 *)	B0	16

Connection of S4P adaptor
*) connected through 10 k resistor for pin sharing

All DIP switches except **A5 MIC** shall be set to on, PVK40 onboard crystal oscillator (3.2768 MHz) shall be used.

No device shall be present in 25xx or 24xx PVK40 socket when SPINET4PVK kit is used. An SPI memory may be placed in 93xx socket, if desired.

The software is preconfigured to use MAC address A6-12-34-56-78-90 and to obtain IP address from a DHCP server. The MAC address may be set by programming EEPROM addresses 0-5. Static IP configuration may be used by writing IP address to EEPROM addresses 8-11 and netmask to addresses 12-15.

If IP configuration is to be obtained by DHCP, successful lease is indicated by audible signal with rising tone. If there is no sound, make sure there is a DHCP server providing leases to anonymous clients on your network. A triple beep indicates that the server has declined the lease. If you do not have a DHCP server on your network, give a try to DHCP Turbo, a DHCP server for Windows by Weird Solutions Inc. http://www.weird-solutions.com/.

Once the board has an IP address assigned, run `pvkpanel.exe`, click connect and that's it!

Please note that there is no webserver included in current version of the demonstration application, you cannot connect to it using a web browser.

2. COMPILING THE SOURCES

The source code was developed using PICC-18 by HI-TECH-Software and debugged on PIC18F458. However, it shall compile just fine for nearly any microcontroller of PIC18 family with none or a few changes. The package also contains precompiled HEX files for PIC18F4520 and PIC18F4620.

If properly modified, the sources may be successfully compiled using another compiler for a microcontroller of a different family or manufacturer.

The make utility is recommended to control the compilation. A win32 port of GNU make is available as part of unxutils package. A suitable Makefile is provided along with the sources.

3. NOTES ON THE IMPLEMENTATION

This software package includes simplified implementation of network protocol stack as a demonstration of functionality of SPINET module in cooperation with PVK40 development board.

The specifics of the microcontroller environment with rather limited resources were taken into account when implementing the network protocol stack. The followed approach was to implement necessary parts of each protocol while keeping it modular enough for further improvements.

The most limiting factor of microcontroller is probably low amount of RAM, thus incoming packets are not copied from internal buffer of ENC28J60 to local memory immediately upon reception, but rather inspected directly in the place and transferred only on request of particular protocol layer or the application, see network buffer access API of enc28j60 module.

An event driven model with callback functions is used, so that the implementation does not rely on a multithreading environment. Processing of incoming packets and any other background tasks have to be periodically called from the main loop. The sources are commented with an effort to make them self documenting.

Please note that the implementation of the network protocol stack is not aimed to be complete or bug free. Consider it as a sample or a starting point provided for you convenience when developing your network application. As the LGPL license says: It is may be freely used but there is no warranty.

4. THE PROTOCOL STACK

- **ARP**: Reply to ARP request only
- **IP**: Inspection of IP header ignoring additional options, there is no support for reassembly of fragmented packets, which typically do not occur on local networks, fragments are simply dropped.
- **ICMP**: Reply to ICMP echo (ping) is implemented
- **UDP**: The application may register a callback function to listen for incoming datagrams on one or more arbitrary UDP ports
- **DHCP**: A simplified DHCP client which allows for leasing and IP address with proper netmask upon startup is implemented, there is currently no support for periodic renewal of the lease.
- **NBNS**: A simplified NetBIOS name service listening on UDP port 137 which is able to respond to simple query is implemented. This provides with convenient connection from Windows machines by simply using hostname "spinet".

- The demonstration application listens on UDP port 2345 and responds back to the sender upon reception of a datagram by sending a status information about PVK40 peripherals - that is how the Windows counterpart, pvkpanel.exe, communicates with the board.

- TCP: Not implemented! Only a skeleton discarding all incoming TCP packets is currently provided. Because of that no higher protocol relying on TCP (e.g. HTTP) are also not part of the demonstration.

5. LICENSING

The code is licenced under GNU LGPL license and thus may be freely used as a base for further development, see lgpl.txt distributed with the source files.