HOCHSCHULE ZITTAU/GÖRLITZ
University of Applied Sciences

Universal Ethernet to SPI communication module for transmission of control messages and upgrade microcontroller software

DIPLOMA THESIS

Faculty of Electrical Engineering and Computer Sciences
Hochschule Zittau/Görlitz - University of Applied Sciences

Author **Jan Korejtko**

Supervisor: Prof. Dr. rer. nat. Stefan Bischoff
Consultant: Ing. Martin Toman, DI-ELCOM

Zittau, 8. May 2013

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics and Interdisciplinary Studies

Study program: N2612 – Electrical Engineering and Informatics
Study field: 3906T001 – Mechatronics

Universal Ethernet to SPI communication module for transmission of control messages and upgrade microcontroller software

DIPLOMA THESIS

Author: **Jan Korejtko**

Supervisor: Prof. Dr. rer. nat. Stefan Bischoff
Consultant: Ing. Martin Toman, DI-ELCOM

Zittau, 8. May 2013

TASK FROM TECHNICAL UNIVERSITY OF LIBEREC

TASK FROM DI-ELCOM

Declaration

I hereby declare that I have written this work independently and used no others than the stated sources and aids.

The work has not been submitted to any other examination board and has not been published yet.

I was familiar with the fact that my Diploma thesis is fully covered by the Act No. 121/2000 on Copyright, § 60 (school work).

I developed the Diploma thesis independently using the literature and based it on the consultation with the supervisor of the thesis and consultant.

Date 8.5.2013

Signature

Abstract

The aim of this thesis is to analyze and realize the possibility of creating an Ethernet interface controlled via SPI to transfer diagnostic and status data. The first part of the thesis deals with the issue of Ethernet converters, SPI bus and conditions for the establishment of such hardware. Creating software section examines the structure of the data transmitted over Ethernet in detail, a brief description of the created functions and their use and a utility for the PC. The last part of the thesis deals with the realization and implementation of the established communication modules, testing their properties both in the laboratory and in operation and economical and functional evaluation.

The main part of the thesis, which is a program for communication module, is attached to the CD and commented in detail.

CONTENT

Shorttitle list

PC – Personal computer

MCU – Microcontroller unit

SPI – Serial peripheral interface

SCK – Synchronous clock, pin of SPI bus

MISO – Master in slave out, pin of SPI bus

MOSI – Master out slave in, pin of SPI bus

SS – Slave select, pin of SPI bus

CS – Chip select, pin of SPI bus

OS – Operation system

OSI - Open Systems Interconnection

UDP – User diagram protocol

TCP - Transmission Control Protocol

FTP – File transfer protocol

IP – Internet protocol

ARP – Address resolution protocol

MAC – Media access kontrol

CRC – Cyclic redudant control

FDDI – Fiber distributed data interface

FTP – Foiled twisted pair

STP – Shielded twisted pair

UTP – Unshileded twisted pair

I2C – Communication protocol in embeeded systems

JTAG- Joint test action group – interface for virtual debuging

ISP – In system programming

TTL - transistor-transistor-logic

FTP, HTTP, DHCP, DNS or POP3

RS 232 – Serial port interface, using UART or USART

USB – Universal serial bus

USART – Universal synchronou/asynchronous receiver/transmitter

UART – Universal asynchronous receiver transmitter

CAN - Controller Area Network

CPLD - Complex programmable logic device

RAM – Random Access memory

RTC – Real time clock

EEPROM - Electrically Erasable Programmable Read-Only Memory

DC –Direct current

GND – Ground potencial

IO – input output

INTRODUCTION

After writing up the bachelor thesis „Independent acquisition and recording of information about engine speed and vehicle speed“ in which I worked with microprocessors of series 8051 and assembler programming langure, I was asked by the opponent of my work why I did not use a higher programming language for microprocessor software development. For this reason I chose to write my Master thesis in the C programming language and for this programming language optimized RISC microprocessor ATmega. The topic was chosen on the basis of cooperation with Dako OJC, a developer of braking systems for rail vehicles and the contracting company DI-ELCOM Ltd. .

The initial idea of the outcome of this thesis was to create a simple communication module based on the converter Lantronix Xport which I used in my master's project "Connection Between integrated Ethernet server XPort and ATmega Microcontrollers" which will serve as the master control system of the PC via Ethernet. Given that the thesis was created in cooperation with DI-ELCOM Ltd. and it was the development of an entirely new device, which will be newly implemented in almost the entire range of the company, it was necessary to adapt the communication module hardware solution so that economically competitive. Because Ethernet converter Xport costs about € 75 its use would greatly increase the production cost of the device. The aim of the thesis was therefore the development of the communication module, the hardware, implementing communication protocols, and the development of a simple utility application for PC. Given the breadth of the range of the company for which the thesis is worked up the topic is focused on the control units of tram cars.

I chose CodeVision AVR as a development environment for microprocessor, which has many of the features implemented in their own libraries. JTAG ICE clone of own production was used to program the processor. The JTAG programmer is not fully supported in CodeVision AVR, that is why translated program was loaded via environment Atmel AVR Studio 5 into the processor.

The Progress of the development of the programs is described only verbally, a detailed description is given directly in the source code in the comments and flow charts. All software, programs, datasheets, schemes and negatives, which were necessary for the thesis developement to be created, are included on the CD.

1. SUMMARY OF CURRENT SITUATION

Because the issue of steering of rail vehicles places high demands on safety, the system is hierarchically divided into superior and subordinated systems as can be seen from Figure [1], detail scheme is in the annex [1]. The part of the maintenance management systems is their service and diagnostics based on indentified error messages. It is necessary to diagnose each system separately at present using RS232 link or CAN bus which is also the main medium of communication throughout the vehicle or in some new systems via Ethernet. This divergence of diagnostic processes increases the time required for maintenance and service. With the development of Ethernet devices, simplifying operation and the trend was thought to centralize diagnostic point for a tram through an Ethernet switch. As the new support systems of rail vehicles such as modul for control of ilumination, air conditioning units etc. are equipped with an Ethernet interface. The new effort if possible is therefore to equip all equipment and vehicle diagnostics with this interface and perform the diagnostics from a single point using a PC. Given that today's Ethernet components are commonly equipped with wireless technology there would not be a need to use a cable connection with a PC in this case.

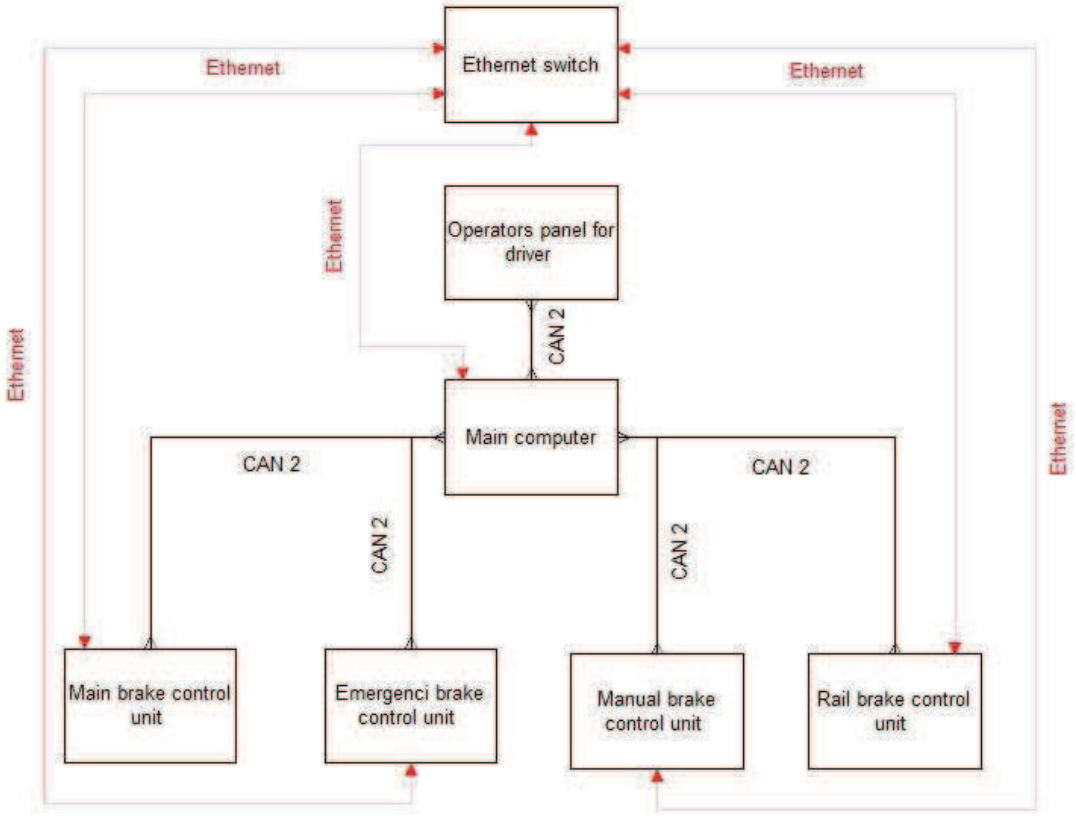


Figure 1: A topological diagram of the portion of the tram electrical equipment

This thesis was developed primarily for the use on multi-tab control systems of railed vehicles. This card was used for debugging and testing the resulting device. Since it was a newly developed control system, hardware of the card was not available from the beginning. Therefore it was necessary to study its functions and elements theoretically. A development kit was made during the work on this thesis to simulate cards on which the program of the communication module was tested. Its functionality was subsequently verified with the prototype of the measuring card and after that even druting the trial operation on the railed vehicle.The multifunction card is equipped with 32bit DSP (digital signal processor) TEXAS TMS320F28335 processor operating at the frequency 150 Mhz. The processor uses a 3.3 V logic. Twelve digital and twelve analog pins of the processor are used as inputs and outputs for measurement connected via isolating elements (impedance, optocouplers) due to the transfer of the signals and the protection against overvoltage damage. Other pins are used to connect RTC hours, 2xRAM for measured data and error conditions and one megabyte I2C FLASH RAM to store program settings. RAMs are supplied by 50F supercap for backup the data. Programmable logic array type CPLDs from Altera is used as the supporting circuitry, which can be used in the processing of fast processes and to protect the equipment when unexpected conditions and critical errors occurs. SPI open collector circuit for switching larger loads is included also. Two CAN buses are used to communicate with other systems, one bus RS 232 and as the last possibility can be used so-called "MULTICHANEL BUFFER" which is the processor interface that can be programmatically set to communicate over different communication protocols. The SPI interface mode is used to control the communication module developed in my work. The picture of the prototype of the controlling card can be seen in the Figure [2]. Multifunctional card will be used for the control of the frequency converter of the motors, control of the rectifiers of the traction and auxiliary drives of the railed vehicle, as well as the control systems for the electric locomotives and brake hydraulic control systems of the trams.

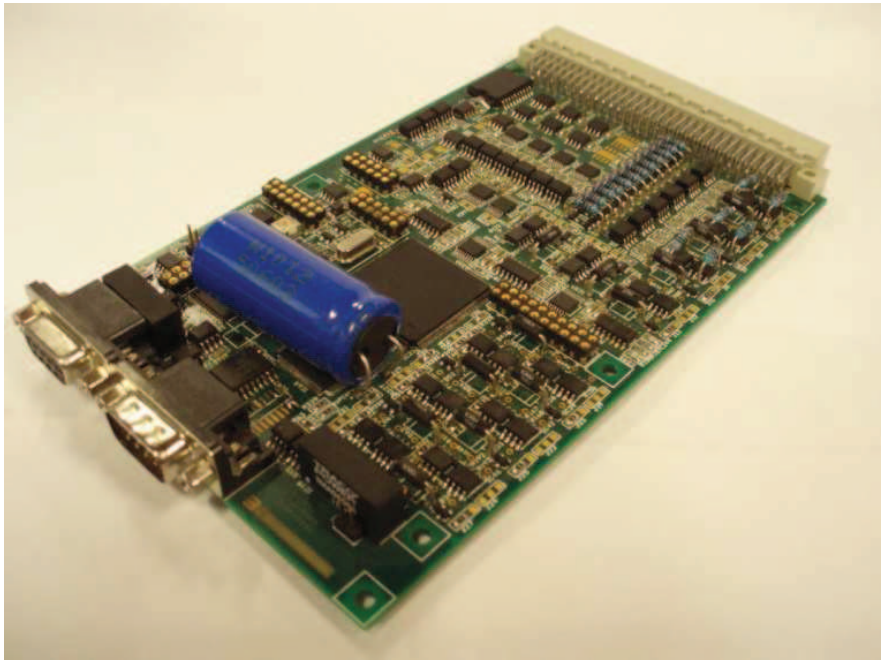


Figure 2: Prototype of DI-ELCOM control card

1.1 Conditions

Hardware and software requirements and conditions of the communication module were determined by DI-ELCOM Ltd. The basic conditions for the development of hardware are the SPI data transfer, logical level of 3.3 V or 5 V, temperature range -30 to +75 ° C, a simplicity of the control of the communication with the superior system versatility and low manufacturing costs.

1.2 Ethernet converters

Due to the development of the Ethernet devices there is approximately 20 Ethernet converters on different communication protocols available in the market in the form of ready-made modules or chips.

Ready industrial Ethernet converter modules, which produces i.e. Papouch Ltd., are very easy to use and are widely programmable interface. But even if they meet most of the requirements for the creation of the thesis and are designed for industrial use their cost exceeds approximately 110€, a reasonable limit for their use in the application. For instance

13 systems are used, for which could be diagnostics via Ethernet applicated in the tram Škoda 15T which is placed in service in Prague.

Separate controllers are used a lot, but due to the complexity of the connections (many outlets) are not very suitable for less demanding or amateur applications, however they are included in some projects as Ethernut 1 (ATmega128 + RTL8019) Ethernut 2 (ATmega128 + LAN91C111) or " Ethernut compatible "Charon 2 (ATmega128 + RTL8019).

Name	Interface	Ethernet verison
Realtek RTL8019	parallel	10Base-T
Crystal LAN CS8900A	parallel	10Base-T,
SMCS LAN91C111	parallel	10/100Base-TX
Microchip ENC28J60	SPI	10Base-T,
Connect One Nano Socket	UART,SPI,USB	10/100Base-TX

Figure 3: Examples of ethernet convertes modules

As can be seen from the table attached on figure [3], converters are equipped with a variety of communication peripherals. The most common is the parallel bus that is not the current standard and therefore the controllers with this bus were not contemplated for use in the thesis.

A converter ENC28J60 was chosen for the purpose of this thesis. Following characteristics lead to this choice: SPI interface, temperature stability, price. Finally, it is essential that the chip is still produced (production of the RTL8019 chip is limited and will be ended soon).

Due to defined conditions the use of the ready-made modules, which are commercially available, is excluded. They do not meet the price, the availability of the communication interface and some even the temperature ranges. For this reason it was decided to develop an own hardware that will use the converter ENC28J60. When using the Ethernet controller will be the conditions set out in the chapter „Conditons“ met.

1.3 SPI bus

Compliance with the conditions of the thesis, and selected Ethernet controllers required the use of two SPI bus. Given that the above bus is in the generally available MCU implemented only once and in the chapter „SOFTWARE“ is worked with it to its lowest functions it was necessary to know in detail its structure and function.

Serial duplex peripheral interface known as SPI can be used to communicate with the memories, converters, displays and other integrated circuits. Communication is performed by four wires attached to the individual integrated circuits, common bus with three wires and one wire for addressing a single integrated circuit. The second wire is used for the control clock signal, the third is used for sending data and the fourth to receive data. The device on the SPI bus, divided into a MASTER type (control) and SLAVE type (subordinate). MASTER is a device that determines to which device it will communicate by setting the address pin to a logic zero (pin marked by the slave, SS or CS). As long as there is logic zero on this pin, receiving and transmitting data shall be available. Another task of the control device is controlling of the communication by the clock signal.The task of the slave is waiting for a logic zero input on SS and transmitting the required data due to the clock signal. Interface pins are called: SCLK (the clock signal), MOSI (Output for the master, input for the slave), MISO (input for the master, output for the slave) and SS (used for decision with which device communication will be held).

Values of SPI protokol logic levels can be combined with so-called modes. The mode of communication determines the moment when the data is ready for fading or sending an interpretation of logic level of each signal. Schema of the communication modes setup is shown in Figure [4].

There is one SPI bus between the host system (measuring card) and the communication module configured as SLAVE and the second bus between ATmega-168 and ENC28J60 converter where the ATmega is the MASTER. Block schema is shown in Figure [5].

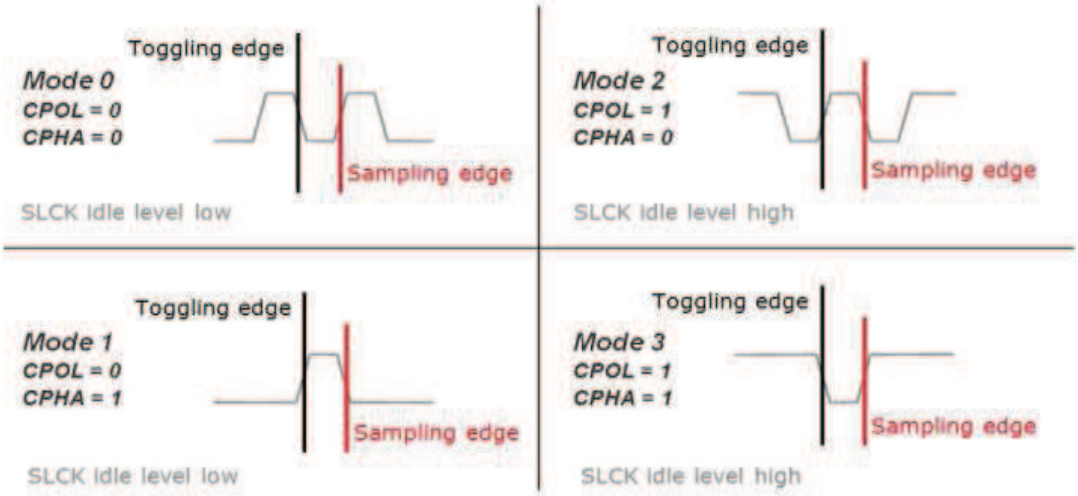


Figure 4: Modes of SPI, source literature [25]

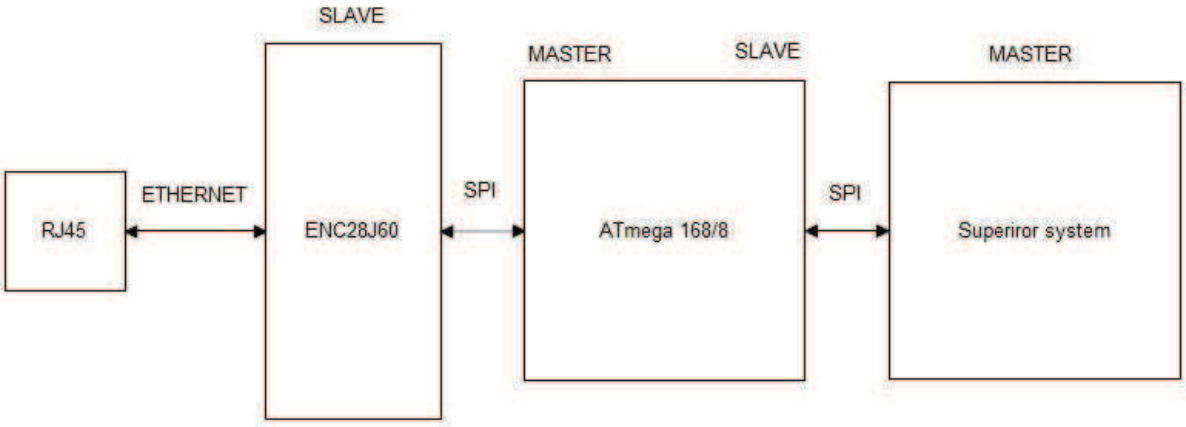


Figure 5: Block schema of connection of the module

2 REALIZATION OF HARDWARE

2.1. Microcontroller ATMEGA 168

Given that the selected Ethernet converter is able to provide only the service of the link layer only, other layers of the OSI model need to be created externally. If these functions should be provided by the direct supervisor system, it would be at that moment excessively busy. For this reason there is one-chip RISC microprocessor of AVR line inserted between the superior system and the converter ENC28J60. That condition of communication with the host system via an SPI interface and via the same interface to Ethernet converter should be met, it is necessary for the processor to have 2 SPI interface. General features of the SPI interface are shown in the chapter „SPI interface“. Because such a single-chip processor is not completely common, or of the inappropriate cost, the AT MEGA-168 Atmel (hereinafter MCU) was selected. This MCU is equipped with one classic SPI interface and the second SPI interface which is based on the USART that is set in SPI mode. This interface can be used only in the master mode. Performance, number of I/O ports can be operated only at 3.3 V satisfies the conditions. Another advantage is the internal oscillator, therefore there is no need to use a crystal.

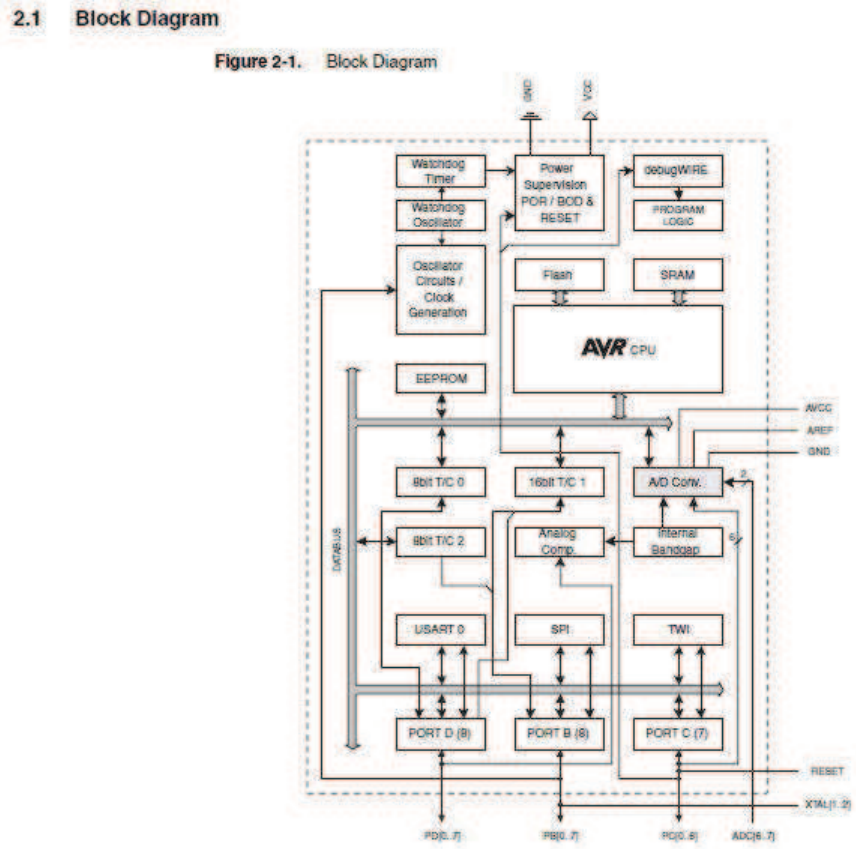


Figure 6: Block diagram of ATmega 168, source PDF literature [3]

The task of the MCU is to process the data received from the ENC28J60, save assessed packets for the superior system or to respond to commands used to control the Ethernet communication. In addition it, must provide information about the stored packets during the intervention of a superior system, pass them and send the data if required. Block diagram of the plugging of the MCU ATmega-168 is shown in the Figure [6].

2.2. ENC28J60 Ethernet converter

One of the products of Microchip ENC28J60 was chosen from the available single-chip converters. The converter is designed to transfer data at the speed of 10Mbit, specifically by the standard 802.3. Electrical and thermal characteristics correspond to the input task, the temperature range -30 to +75 ° C, operating voltage of 3.3 V. The converter is controlled and programmed via SPI interface and is able to take one-byte instruction. It provides a link layer, see the chapter ETHERTNET, for communication via Ethernet. The control register memory of ENC20J60 is divided into several areas, Figure [7]. Area of the control register is divided into four banks.

Bank 0 Address	Name	Bank 1 Address	Name	Bank 2 Address	Name	Bank 3 Address	Name
00h	ERDPTL	00h	EHT0	00h	MACON1	00h	MAADR5
01h	ERDPTH	01h	EHT1	01h	Reserved	01h	MAADR6
02h	EWRPTL	02h	EHT2	02h	MACON3	02h	MAADR3
03h	EWRPTH	03h	EHT3	03h	MACON4	03h	MAADR4
04h	ETXSTL	04h	EHT4	04h	MABBIPG	04h	MAADR1
05h	ETXSTH	05h	EHT5	05h	—	05h	MAADR2
06h	ETXNDL	06h	EHT6	06h	MAIPGL	06h	EBSTSD
07h	ETXNDH	07h	EHT7	07h	MAIPGH	07h	EBSTCON
08h	ERXSTL	08h	EPMM0	08h	MACLCON1	08h	EBSTCSL
09h	ERXSTH	09h	EPMM1	09h	MACLCON2	09h	EBSTCSH
0Ah	ERXNDL	0Ah	EPMM2	0Ah	MAMXFLL	0Ah	MISTAT
0Bh	ERXNDH	0Bh	EPMM3	0Bh	MAMXFLH	0Bh	—
0Ch	ERXRDPTL	0Ch	EPMM4	0Ch	Reserved	0Ch	—
0Dh	ERXRDPTH	0Dh	EPMM5	0Dh	Reserved	0Dh	—
0Eh	ERXWRPTL	0Eh	EPMM6	0Eh	Reserved	0Eh	—
0Fh	ERXWRPTH	0Fh	EPMM7	0Fh	—	0Fh	—
10h	EDMASTL	10h	EPMCSL	10h	Reserved	10h	—
11h	EDMASTH	11h	EPMCSH	11h	Reserved	11h	—
12h	EDMANDL	12h	—	12h	MICMD	12h	EREVID
13h	EDMANDH	13h	—	13h	—	13h	—
14h	EDMADSTL	14h	EPMOL	14h	MIREGADR	14h	—
15h	EDMADSTH	15h	EPMOH	15h	Reserved	15h	ECOCON
16h	EDMACSL	16h	Reserved	16h	MIWRL	16h	Reserved
17h	EDMACSH	17h	Reserved	17h	MIWRH	17h	EFLOCON
18h	—	18h	ERXFCON	18h	MIRDL	18h	EPAUSL
19h	—	19h	EPKTCNT	19h	MIRDH	19h	EPAUSH
1Ah	Reserved	1Ah	Reserved	1Ah	Reserved	1Ah	Reserved
1Bh	EIE	1Bh	EIE	1Bh	EIE	1Bh	EIE
1Ch	EIR	1Ch	EIR	1Ch	EIR	1Ch	EIR
1Dh	ESTAT	1Dh	ESTAT	1Dh	ESTAT	1Dh	ESTAT
1Eh	ECON2	1Eh	ECON2	1Eh	ECON2	1Eh	ECON2
1Fh	ECON1	1Fh	ECON1	1Fh	ECON1	1Fh	ECON1

Figure 7: Table of ENC28J60 registers, source literature PDF [5]

The each bank contains 25 unique registers and 5 that are common to all banks. Unique registers are used to internally set the transmitter such as the authorization of the filters, setting of the Full-duplex or the half-duplex mode, MAC addresses etc. The common

register holds the pointer for the currently selected bank, the state of the chip, handling of the interruption and operation of sending data.

The recommended connection diagram given in Figure [8] was followed when designing the circuit for the Ethernet interfaces. Because Ethernet connector contains the necessary coils and high voltage decoupling capacitor, it was needed to connect only resistors, inductor and two capacitors in the circuit. Internal wiring of Ethernet connector is shown in Figure [9].

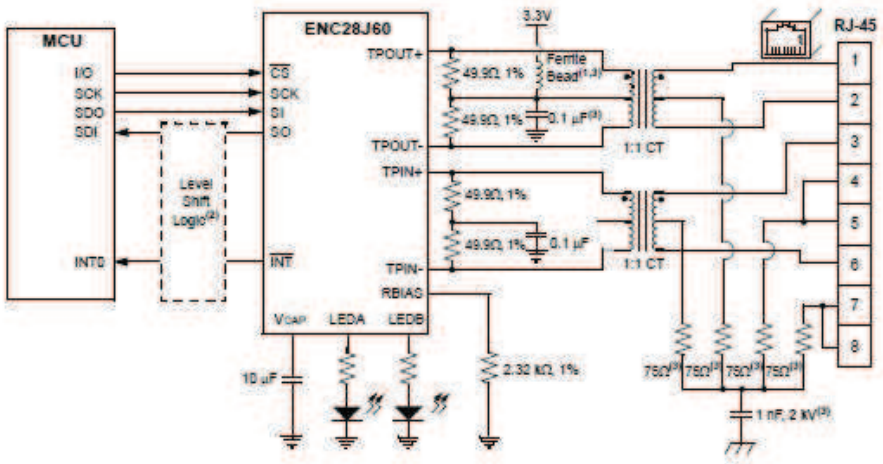


Figure 8: Recomended connection of ENC28J60 converter, source literature PDF [5]

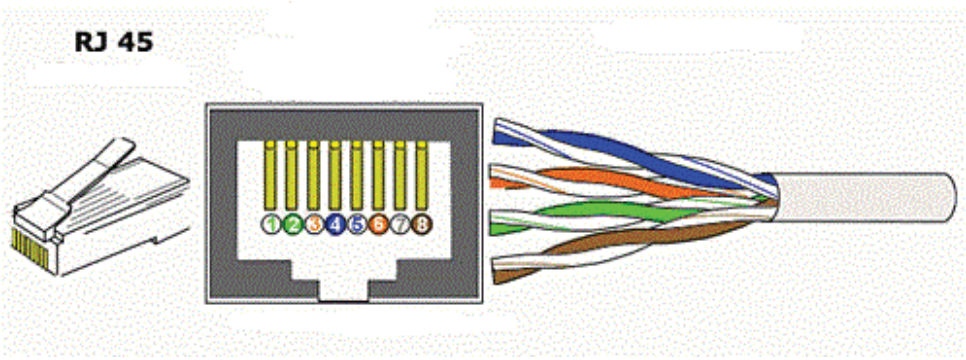


Figure 9: Wiring of RJ45 connector source [27]

There is a 25MHz crystal connected between the pins OSC1 and OSC2, required by the converter. The controller also requires a capacitor with a capacity of 10 uF and a resistor of the value of 2.32 kohm - these components are necessary for the proper functioning of the internal voltage regulator and the ENC28J60 transmitter.

It is necessary to assign each device a unique identifier of the network devices that use different protocols second (connection) OSI layers for the use of the converter in the

Ethernet network. It is the so-called MAC address of the device. It is assigned to the network card or device immediately during its production (for older cards it is stored directly in the EEPROM memory) and therefore it is sometimes called the physical address, but it can be nowadays changed in modern cards retrospectively. Ethernet MAC address consists of 48 bits, and according to the standard should be written as three groups of four hexadecimal digits (eg 0128.6867.89ab), more often it is written as six two-digit hexadecimal numbers separated by dashes or colons (eg 01-28-48-90-89-ab or 01:28:48:90:89: ab). Each hexadecimal pairs is converted to a binary number during the conversion. The MAC address assigned by the manufacturer is always globally unique. It is divided into two halves in terms of allocation. The manufacturer must apply the main part for the first half of the address space. This half is the same for all cards of the same manufacturer (or at least large groups of cards, the big manufacturers have several values for the first half). The manufacturer then assignes a unique value for the second half of the address forevery card or device made. This uniqueness makes managing of local area networks very easy - a new card can be connected and rely on that it will clearly identified. Due to the fact that the transmitter ENC28J60 has no MAC address assigned to from the factory it is neccessary to use the address defined by the user. It is possible to make up the MAC address and test that if it is unique on the network, or you can use the MAC address of discarded old network card which is guaranteed by its uniqueness.

Given that the module will be used only in a small local network without access to the internet, it is not necessary to think about the originality of the MAC addresses of each module, hence the addresses were devised. If it is necessary to change the default MAC address, it is possible to do this either from the master system or by the user after connecting the module to the PC. When defining the MAC address by the user, he must avoid MAC addresses that have special meaning, such as omnidirectional (broadcast) address identifying all connected devices. Its addressed packet will be accepted by all devices on the local network. Broadcast address has all ones (ff: ff: ff: ff: ff: ff). The group MAC address (multicast) is a group of connected devices. It will be taken by all devices on the local network that has been configured as members of the group (typically an application request to join the group and the card will then receive packets with a given group address). Group addresses have in the least significant bits of the first byte number one.

2.3. Ethernet transformators and connectors

Metallic transmission medium of the Ethernet is separated from the individual devices connected in by galvanic transformers. In practice it is a circuit that integrates transformers and other elements. It is a part of the Ethernet controller and an RJ45 connector used for connection to the Ethernet network. For simplicity of the device design RJ45 connectors with isolating transformers started to be produced. They are commonly called MagJacks but this name is a trade mark registered by Bel Fuse Corporation which began producing them as one of the first. Currently there is a number of these connectors from Chinese producers on the market who are usually calling them by an obligate description - RJ45 modular jack with pulse transformers etc. The typical appearance of the connector and the internal wiring is indicated in Figure [10].

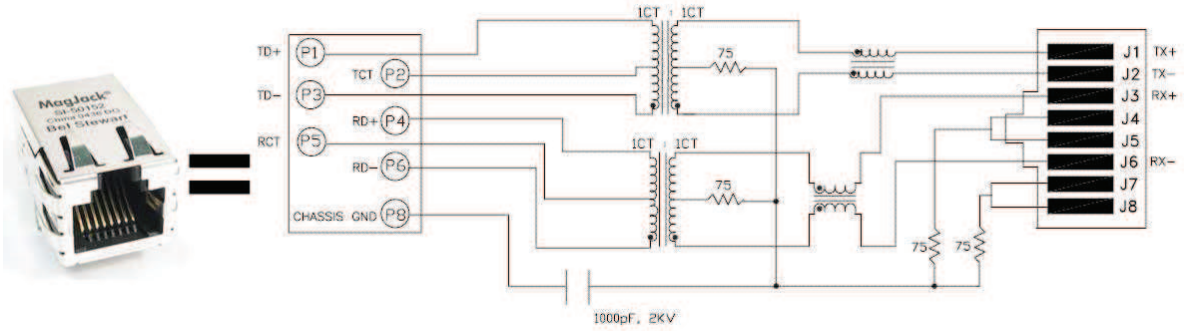


Figure 10: Magjack with internal wiring diagram, source literature [20]

MagJack is typically a combination of connetion involving more elements. These include RJ45 connector, impedantly customized magnetics transformers or other specific standards for the network 10Base-T, 100Base TX and 1000Base-T. Some are adapted for the possibility of power charging of the devices by the data cables - Power Over Ethernet (PoE). Further standard includes resistors, high voltage equalizers and filters that suppress the consistent interference. Signaling diodes may be integrated in the housing optionally. Using of MagJack simplifies the design and reduces the size of the resulting device. There is a wide variety of these connector types on the market, and it is important to respect the distribution of individual outlets which is not the same for all models when designing the connection with Ethernet.

To make a connection of the communication module to an active network element by UTP cable possible, was necessary to implement RJ45 socket with in the communication module. Then also needed to ensure electrical isolation from the Ethernet physical layer to protect the terminal device in the case of an unusual strain on the twisted pair cable. Galvanic

isolation is performed by isolating transformers for Ethernet. Single MagJack was used during the construction of the module to save space and minimalization of the size.

2.4. Programmer

A programmer was designed by literature [24] for the purposes of the thesis. This is a clone of the original programmer JTAG ICE Atmel. The programmer is compatible with the full range of Atmel products and can be switched into a mode JTAG or ISP programming. It is supported by both the environment AVRStudio under the J-TAG and environment AVROSP-II in ISP mode. Designed programmer is shown in Figure [11].

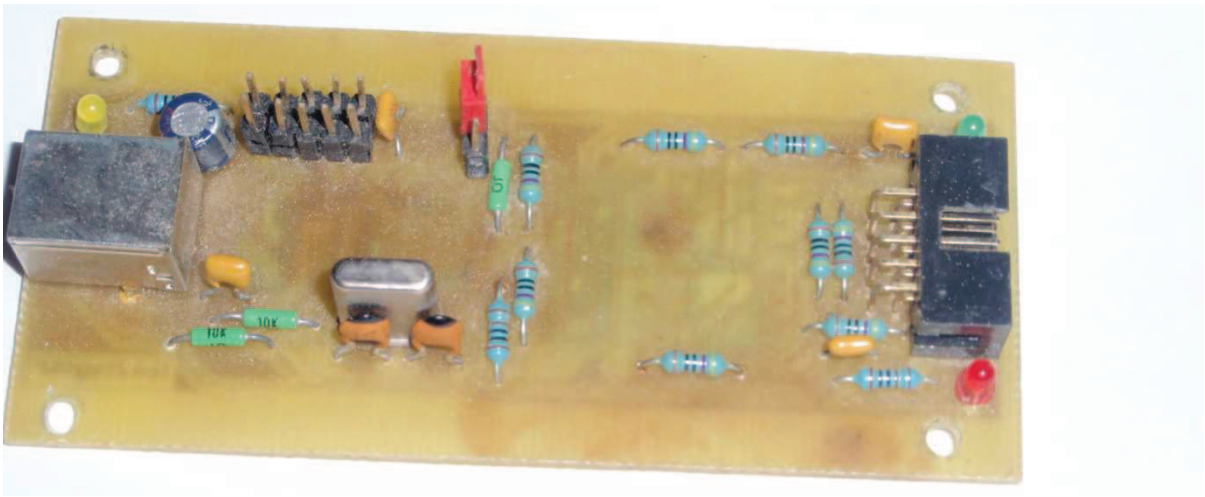


Figure 11: ISP/JTAG programmer

2.5. Development kit

According to previous chapters the development kit was compiled for debugging the program of the converter, simulation of the transmitted and received data and verification of the electrical and thermal characteristics of the whole module. Kit is supplied with 5V, which are changed through a simple step-down converter with LM317 to the required 3,3 V. The main part of the development kit are two MCU ATmeg-168, which are interconnected by the SPI interface. The first one represents a superior system which is connected via USART to the PC where it is possible to send simple commands like from the superior system. The second MCU provides higher layers of the OSI, communicates with ENC28J60 which is connected via a socket of 2x5pins. There is a jumper switch that can disconnect the processor from the power supply for each of the processors. It is because those both processors are connected by the SPI interface, through which they are simultaneously both

programmed. It is therefore necessary to be able to disconnect each processor separately. Furthermore, the development kit board contains two LCD display 2x16 characters that are connected to the processors and they also listing the processor state when debugging. A scheme of the development kit is shown in Figure [12].

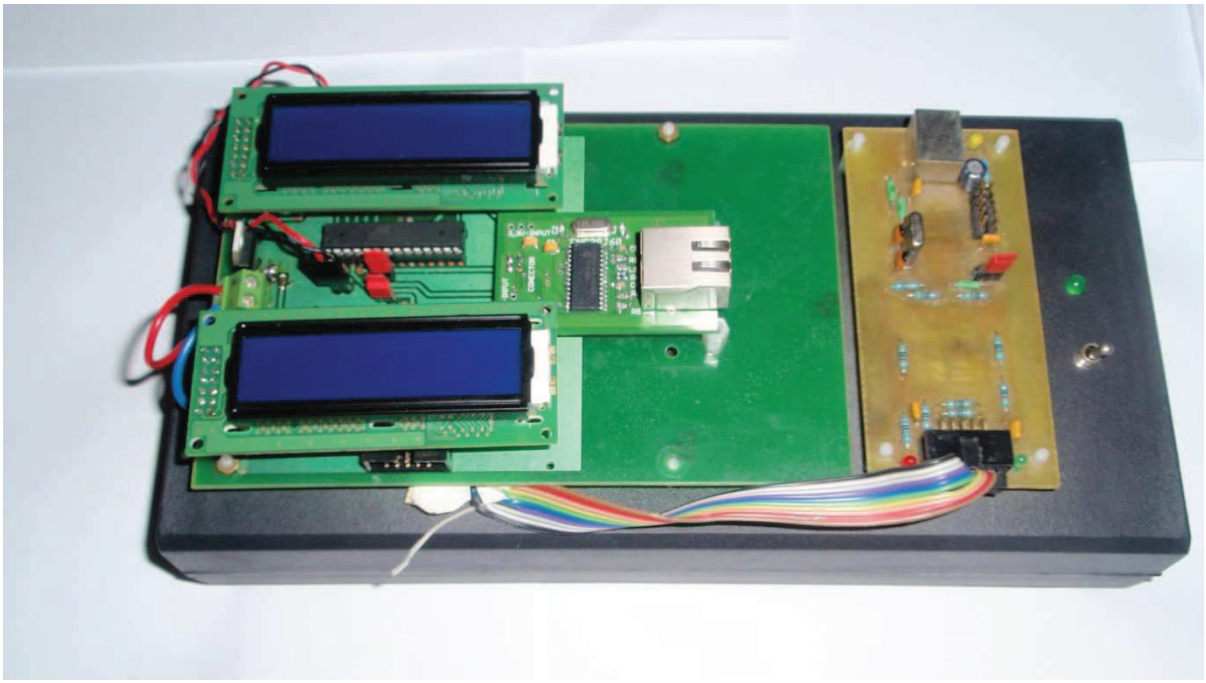


Figure 12: Development kit

MCU simulation superior system is connected to the PC via ASIX cable converting RS-232 levels with 0-5V levels for USB. With the controls on the transmitter the PC virtual COM portis installed through which it is possible to send and accept converted messages. Given that the used conversion cable maintains the polarity of RS-232 levels (logical 1 = 0V log.0 = 5V). Own converter based on FTDI chip-232RL was designed which is used in the cable ASIX with embedded optical isolator connected. So that inverts the output and the input signal to the chip so that the logic level on the side of the MCU are in conformity. A scheme of the conversion of the signals is shown in Figure [13]. The Terminal freeware program was used to communicate through the virtual port that allows complete service of the virtual port, including text and binary display of the incoming and outgoing messages. The program Terminal is a part of the attached CD.

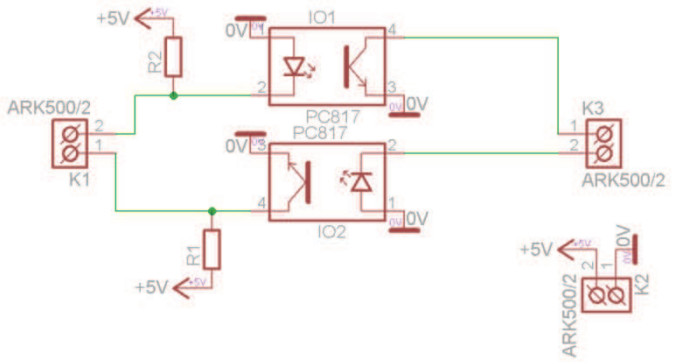


Figure 13: Signal converter schema

The actual development kit was used to develop and debug software for the later constructed individual communication module. There was a problem with using the hardware SPI microcontroller ATmega-168 during the software development, in communication with the transmitter ENC28J60. The hardware controller MCU, which provides a clock signal and the physical transfer of data stored in the register SPDR, is designed for the transmission of data of one byte while handles CS data signal by itself. In order to exploit the potential of the speed and versatility of the communication with the chip ENC28J60 it is necessary to accept some 16 bit messages to 8-bit instruction as a reaction, such as reading of the registers etc. For this reason it was refrained from using ATmega-168 microprocessor which has the advantage of being able to use the two SPI interfaces before the construction of the final module. Instead, after consideration, the MCU ATmega-8 was chosen, which has the same configuration of the IO port, so there was no need to change the design of the development kit, it has smaller size in the SMD case used in the final module and last but not least it is cheaper. The SPI interface to communicate with the ENC28J60 was created using the software of the library of functions of own designin chapter „Communication with ENC28J60 converter“. The procedure of developing of the kit components with dimensions in the standard DIL size, and the wiring was proven to be correct, construction of the final module without additional shortcomings seem to be right.

2.6. Communication module

After tuning the basic functions of the Ethernet communication on the development kit, it was simplified according to the wiring the diagram in annex [3]. There are 4-pins of the SPI bus wired from the module to communicate with the host system, then the 2-pin for power supply of 5V and the last one is the 3 pin connector that is primarily intended for

programming the MCU. There are wired GND, 3.3 V and RESET signal on it. A Self-supply is done via 5V stabilizer KF33BD because during the development it was found that when sending a packet the current consumption of the current module is up to 250mA. There is no 3.3V branch available or designed for such consumption from the control board so the power module is made from a 5V branch that is current-stable. The main module is shown in the Figure [14]. The indisputable advantage of the Ethernet networks is the ability to connect via wifi. Commercially available Ethernet switches eligible for centralizing diagnostic points have this feature. This brings an opportunity to make a connection with the switch and communication modules without opening the place where a switch in the railed vehicle is stored so an Ethernet cable could be connected.

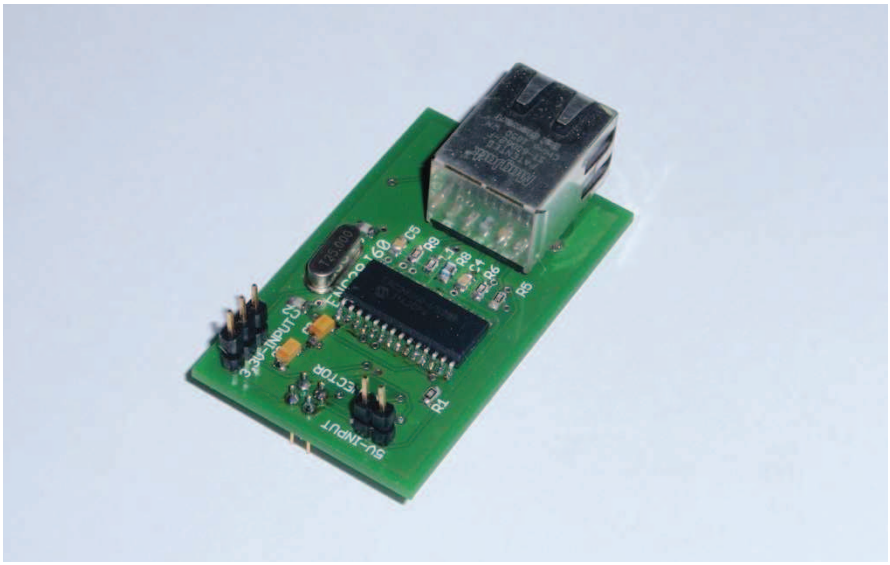


Figure 14: Communication modul

The resulting module consists of only:

1. converter ENC28J60
2. microcontroller ATMEGA 8
3. MagJack connector
4. I2C EEPROM 24C64
5. voltage regulator KF33BD
6. passive components

EEPROM memory has been added to the diagram for the possibility of storing larger amounts of data such as software updates from the PC terminal.

3. SOFTWARE

3.1. Ethernet

As already mentioned, Ethernet will serve to connect the system. It is a technology that solves the problems of physical and network layer protocols TCP / IP and ISO / OSI. By the developement the bus topology became a Star (or type of a tree), where hub (HUB) joint the center star. This is just the replacement of the bus topology. Its function is to pass on the received data to all connected participants. Later hubs were replaced by switches (switches). These devices can store data received and from it alone determine which channel to use to send data only to the designated addressee. This eliminates the need to wait for the release of the transmission medium and the entire network is therefore more permeable. As the media for the Ethernet is now used primarily optical fiber and twisted pair. The advantage of the optical lines are long distance transmission and immunity to electromagnetic interference. At shorter distances, unless the conduct is exposed to strong electromagnetic interference from power equipment, it is cheaper to use metallic cable, such as the twisted pair mentioned. Unshielded cables UTP or shielded STP which are more resistent to interference are used. Ethernet is now characterized as a serial, full duplex communication. For four pair cables for 100Base-TX (Fast Ethernet) are used only two pairs and pins 1,2,3 and 6. Others remain unused. Use the remaining pins comes with 1000Base types. The remaining two pairs for 100Base-TX are also used for the DC power equipment on the network, such as switches, routers, repeaters, etc. There will be used a UTP cable with RJ-45 wired according to T568B standard 10/100BaseT for the physical connectionin this module. Orange and white-orange wire is used to send data to a green-white-green to receive data. The rest, as has been already mentioned, are not used.

3.2. Ethernet data transmission

It was necessary to understand the structure of Ethernet communication and interpretation of data contained in the packet in detail for elaboration of the diploma thesis. It is divided into four layers, which are hierarchical. This means that each layer uses the services provided by the lower layer. At the same time, however, offers its services to the layer above. This architecture allows a replacement of the individual protocols. Communication takes place between the same two layers, which contain communication protocols. Layers and individual services they provide are shown graphically in the Figure [15].

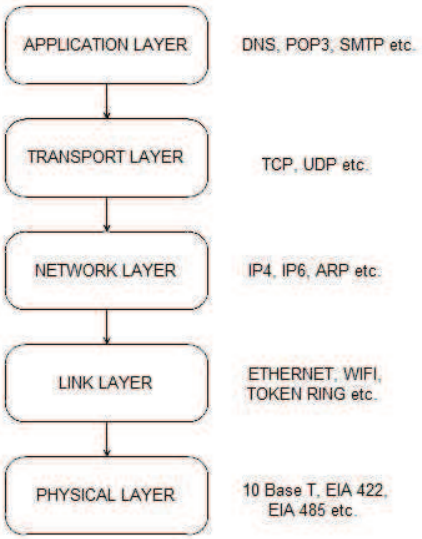


Figure 15: OSI layer distribution

Application layer:

The highest layer is the application layer. It represents the data transmitted over the network which are interpreted by the program. The most common protocols for the data transfer are protocols FTP, HTTP, DHCP, DNS or POP3. The top layer is not used during the communication of the developed module. Data are interpreted using PC applications developed in C# and are transmitted in the form of raw data without interpretation application layer protocols

Transport layer:

The second highest layer is a transport layer. It allows applications to address directly with the help of the so-called port. Port is written in the protocol header using two 16-bit numbers which determine the port of the sender and the port of the recipient. Another common parameter is a checksum for the protection of the data received. If the error was detected during transmission, then the data would be requested again or discarded.

Transport layer represents several such protocols which differ in their functions. They can be divided into two groups, connective and non-connective. Linked protocols operate on the principle of request-response. This function ensures 100% delivery of the transmitted data. If the recipient does not respond to the data received by a confirmation message, data are considered as undelivered and sent again. This effectively creates a communication tunnel that maintains the connection until the communication is not terminated. The most known connective transport layer protocol is TCP. Its advantage is, as has been already mentioned,

secured receipt of the transmitted data, it is simultaneously also a disadvantage, because in large networks this form of communication is slow. The second group is non-connective protocols. They only send data to the recipient. The most common non-connective protocol is UDP (User Datagram Protocol). This protocol does not ensure the delivery of data, but it is not bound to a unique IP address. It can be delivered as a broadcast or multicast messages. All users on the network, or group of devices shall receive these reports. Also faster in comparison with TCP. It is mainly used to transmit real time data, to transmit video or audio, when large volumes of data transmitted. However, despite being able to transfer large amounts of data, it is not recommended to fragment data, precisely because of the nature of non-connective protocol. UDP protocol is defined by RFC 768, which is annexed to the CD.

Network layer:

This layer routes data through the network to the remote user. It provides links between the systems which are not directly adjacent. It contains functions that allow you to bridge the different characteristics of technologies in transmission networks. The most commonly used network protocol is IP. Another frequently used protocols are ARP and ICMP which are used to determine the MAC address of the device with a known IP or reporting error messages. They can therefore be used to communicate with network elements, not only with terminal devices. IPv4 is used for sending data or IPv6 which is still not fully extended. In a small network, use just IPv4. The main part of it are the IP addresses which are 32-bit numbers. Two IP addresses are put into the network header – an adress of the receiverand the address of the sender. As it was necessary to understand the IP protocol in detail, it can be implemented into the MCU followed by listing of dwellings IP diagram:

Protocol Version: takes the value 4
Length header: shows in four apartments, the minimum size is 20 bytes,
Type of service: should ensure priority delivery service is not used
Overall length: Specifies the length of the datagram, which can be up to 65535
Datagram identifier: use of the fragmented datagram data indicates that belong together.
Symptoms: This is a notification enable / disable fragmentation and notice that this is the last fragment
TTL: Specifies the lifetime of the network in order to avoid endless spread
Higher protocol layers: the identification number of the parent protocol (TCP = 6, UDP = 11)
Checksum: CRC of IP protocol header

The network interface layer

The last layer is dependent on the used technology of the transfer. TCP / IP or UDP protocols do not deal with it too much and it depends on the technology used in the signal transmission. This layer thus defines what media will transmit data. Transmission can take place wirelessly (WiFi), metallic (UTP cable) or optically (fiber). Technology for data transfer are for instance: Ethernet, Token Ring, FDDI etc. Each technology has its own communication and add its own header to the data received from the network layer.

The module has to be connected via Ethernet either via WiFi or via cables. Data will be routed through IP and for them to follow the UDP protocol in which raw data are contained. This configuration was approached because of the speed of the data transfer and the assumption of sending small frames (up to 255B of raw data). Another advantage of UDP is the ability to send messages of the type Broadcast without clearly defined recipient. This is used to identify devices connected to the network. The structure of the packet containing the IP and UDP are analyzed graphically in the Figure [16]. This structure was created on the learned standards of the UDP and IP protocols and standards of the Ethernet. These standards are as an anex on CD. For the detailed analysis of the properties of the protocols that are defined in the standards, but not used in the current versions see the chapter „Network trackers“.

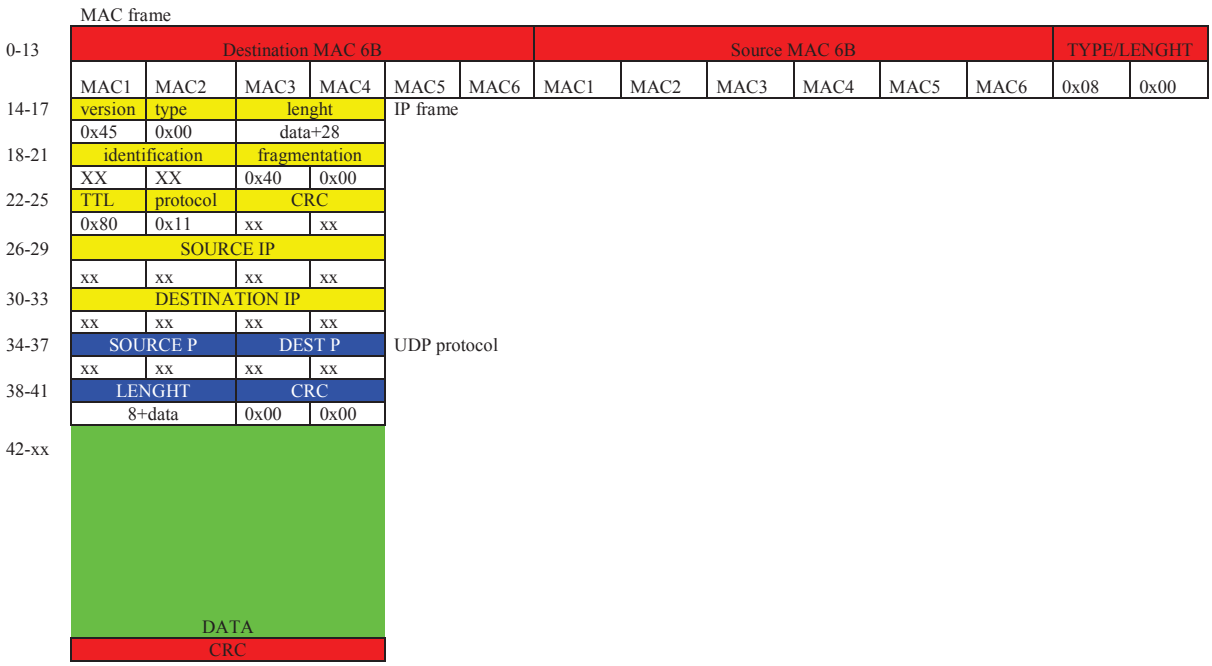


Figure 16: Graphic description of ethernet packet

3.3. Software for communication module

Software communication module must ensure the processing of higher layers of the OSI model, ensure the commmunication and setting of the ENC28J60 circuit and provide information to a host system on its state or the state of the converter ENC28J60. The program was created in the development environment CodevisonAVR that is compatible with ATmega and includes libraries for the basic microcontroller peripherals, unfortunately, it is not compatible with the established JTAG / ISP programmer. The actual machine code was therefore loaded to the MCU from the AVR studio environment. The basic structure of the program is shown in the flow chart in annex [4]. As the flowchart shows the entire program is divided into two parts that are servicing INT0 interrupts and disruption caused by an early communication via SPI. The main loop of the program remains empty and the MCU does not perform any operations in it.

3.3.1. Software for communicating with the master system

The microprocessor communicates with the host system via a simple protocol that was created on the basis of agreement with DI-ELCOM Ltd. and allows basic operation and setup of the converter ENC28J60 and operating of the data that are currently available to the superior system. The communication protocol is shown in the table in the Figure [17].

ID prikazu(1byte)	Data(byte)	Popis
2	1	State of modul [1=Ready;2=Bussy;3=Data Rx;4=Data Rx Full;5=Error]
3	Max 255	Data to send, end of data is set by SS seting to logic one
4	Max 255	Reading saved raw data, first byte is lenght of it
5	12	Reading MAC, IP and PORT
6	1	Set the promiscuit mode
7	4	Set new IP of modul
8	6	Set new MAC of modul
9	4	Set destination IP
10	6	Set destination MAC
11	2	Set destination PORT
12	0	Send data to ETHERNET
13	0	Reset

Figure 17: Communication protocol

So this communication could be managed independently by the superior system a communication module is set to the Slave mode, which means that it can not initiate communication and must wait for the pulse (input value on pin SS = 0) from the host system. The service of the interruption identifying the ID command received from the host system.

According to the ID statement are necessary instructions made. These instructions are placed in positions which, however, are no longer part of the interruption. Apart from RESET instruction all functions are followed by the reaction of either receiving or sending data. For this reason the call to interrupt must be off at this time. Identification of whether the transmission or reception is complete SPIF flag, which is a bit in the control register MCU ATmega-168.

Function status_SEND

This function is used to send the value of the STATUS variable. A superior system can identify whether there are any commands available for it from Ethernet, or whether the communication module is ready to accept data to be sent to the Ethernet according to this variable. In the event that there is a situation where the communication module only assembles the packet and prepares it for the shipment and superior system would have to send more data would result in a collision of these two packets and the data would be discarded. The principle of this feature is thus that the host system periodically (when not carrying out any substantial activity) asks whether the communicator is present in a control packet. If so superior system states to the state of the active communication and reads the data. If it finds the answer it responds to the sender. If the responding is permanent answering i.e. periodic sending of the actual values, this function is used to determine whether a packet has been sent and is thus treated a possibility if a request was received for the next data is sent without prior already processed.

Function My_IP

If a request to change the IP address of the moduleis received, the module waits to receive 4 bytes with the new IP address. Because of the universality the packet with the request to change the IP address is passed from the PC to the host system and it makes the change. It is for that the user of the PC or separately superior system are able to make changes.

Function My_MAC

Is a function to change the MAC address of the module. The principle is the same as in My_IP, but it receives only 6 bytes and the setting of the MAC address of the module is changed. The MAC address of the module has also to be changed directly in the internal register ENC28J60 converter, where it is registered using the

encCtrlReg_Write (addres, data) from a source ENC_FUNCTIONS that is described in the chapter „Software for communicating with ENC28J60“.

Function Destination_address

This function is used to set the recipient of the addresses of the transmitted packet. The address is entered by the superior system and it is not needed to enter it every time. If i.e. periodic sending data about the state of the measuring cards is required simply it is necessary to enter the recipient's address only once then just restore the data by the command data_toSend. The recipient address is put in in the form PORT, IP, MAC.

Function Reset

In the event that status_SEND function () returns an ERROR value to supervisor it is better to reset the module. First you need to reset the converter ENC28J60 itself. This can be done in software or hardware. The system of the hardware reset by connection PD.3 to the RESET pin of the converter ENC28J60 was used for the purpose of the thesis. According to the datasheet which is annexed as literature [2], it is needed for the reset pulse to last at least 5 ms. It is also necessary to reset the MCU itself. An integrated system WATCHDOG is used to do this. A reset is done by writing WDRF values to the WDE registry, the watchdog timer is started by it and after 15ms reset of the MCU is done.

Function Promiscuit_mode

Promiscuit mode is a special mode of the ENC28J60converter. If this mode is activated, it switches off all filters and all packets arereceived. This feature is reserved for the future or testing use.

Function data_toSend

Raw data to be sent are entered using data_toSend. The maximum length of data to send is 255B. It was established as a sufficient amount of data. Data to be sent are processed directly in the main memory of the processor are sent directly to the transmitter buffer ENC28J60 for processingafter the calculation of the CRC. Functional block diagram data_toSend is shown in Figure [18].

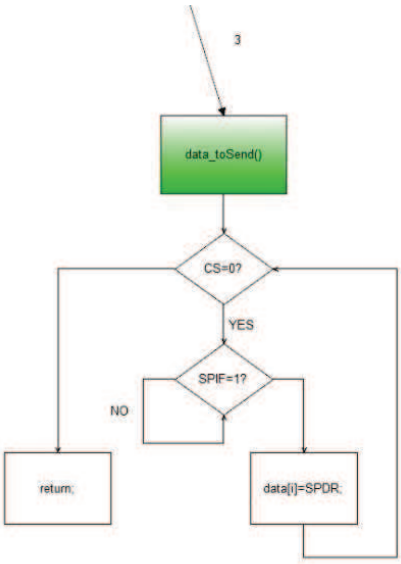


Figure 18: data_toSend function diagram

Function data_Read

This function is used to read the bare instructions received via Ethernet. These instructions are stored as well as the sender's address in the EEPROM on the I2C svěrnici, from which they are read by the MCU module. After receiving the request to read the data MCU sends the first information on the length of the message and then sends the main data. This happens because the MCU acts as the slave in communication and is not able to control the number of clock pulses that are needed to read the data. It is therefore necessary to inform the host system how many bytes are to be read. Functional block diagram data_Read is shown in Figure [19].

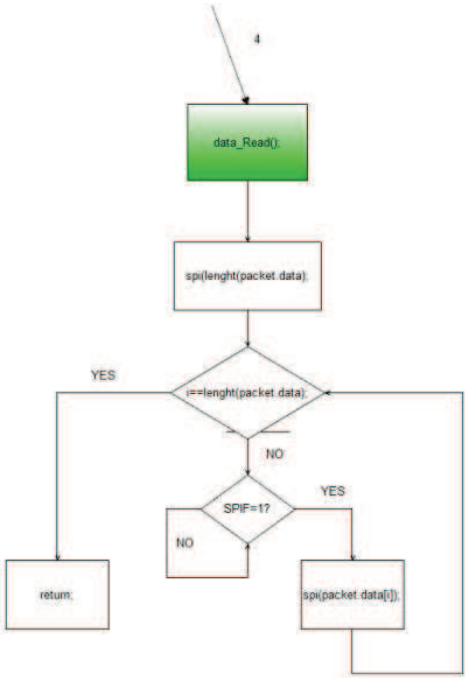


Figure 19: Data_read function diagram

FunctionAddress_read

This function is used to read the addresses of the sender packet by the master system. It is for the case, if the system communicates with the multiple device at once. Therefore, the recipient address is not processed by the module, but by the superior system. The structure of the address is in the form of the MAC address, IP address and Port. Number of the received byte is fixed. Flow chart of the function is in the Figure [20].

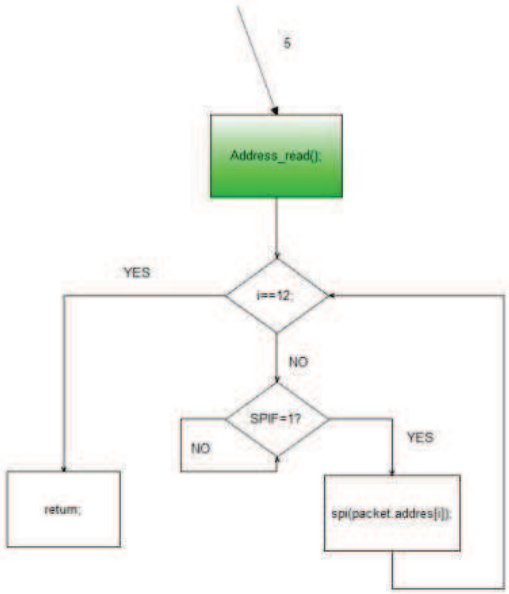


Figure 20: Address_read function diagram

3.3.2. Software for communicating with ENC28J60

MCU communicates with the transmitter via the SPI bus. Compared to the communication with the host system the MCU is in MASTER position. Due to problems with the setting the USART ATmega-168 to SPI MODE0 mode a library that defines the SPI master communication software was created. Further there are defined higher functions of the communication with an ENC28J60 converter, which use USART_SPI library. These functions are used to read and write data to the registers and buffer. These functions are based on the ENC28J60 datasheet. The highest functions of this program are to evaluate the received packets.

Library USART_SPI

This library of function is used to mediate software communication between the converter ENC28J60 and ATmega MCU. There are pins SCK, MISO and MOSIdefined in it. Pin CS is always controlled before initiating communication in higher positions. The clock signal SCK is controlled by software. Communication speed is set approximately to 1 MHz. A picture of the communication by the software SPI from the oscilloscope is in the Figure [21].

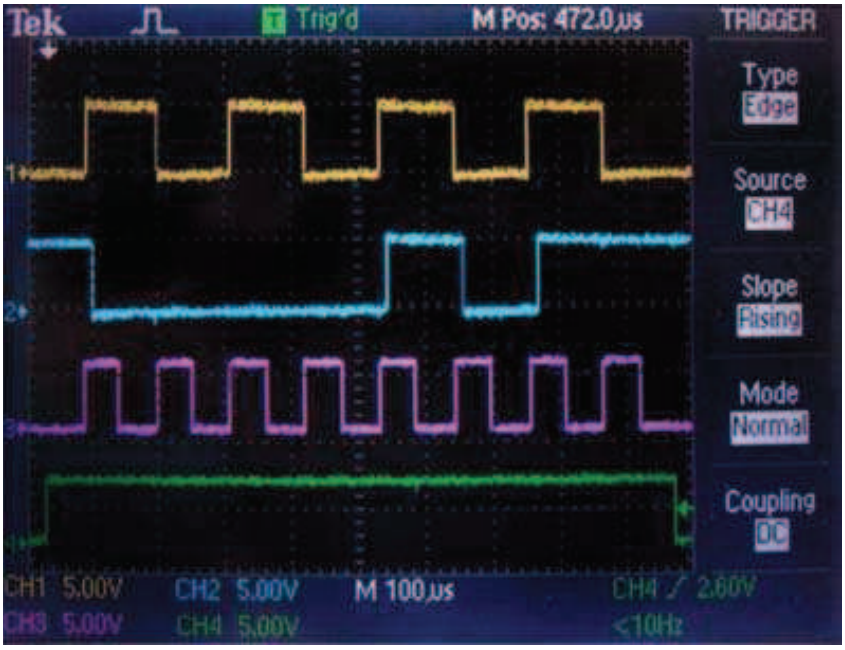


Figure 21: Software SPI communication, green=CS, violet=SCK, blue=MISO, yelow=MOSI

Function SPI_write

It is a function for writing one byte after an SPI. SCK pin sends 8 clock cycles. It is immediately followed by a shift of the bit value of the current byte sent to the MOSI pin. The flowchart of the SPI_write function is shown in the Figure [22]

Function SPI_read

This function mediate reading of one byte from the SPI interface. 8 clock signals are dispatched at the SCK pin, while in the middle of each signal the value of the MISO pin is read and the desired byte is received. Flowchart of the SPI_read function is shown in the Figure [22].

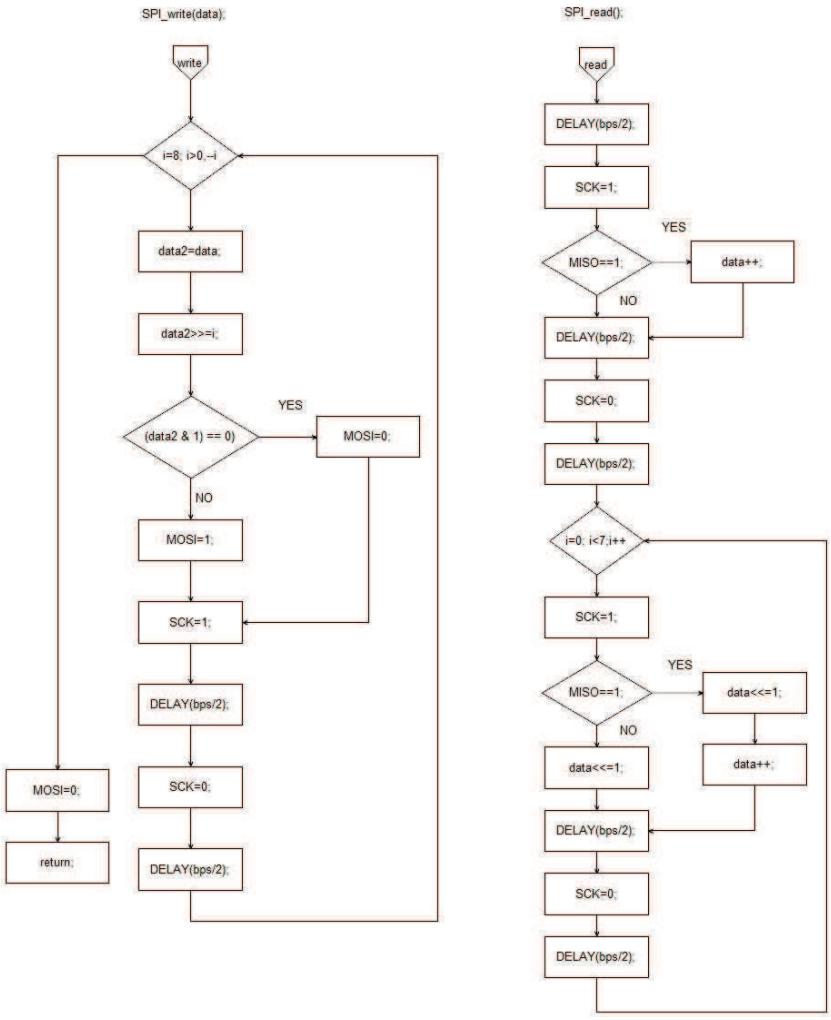


Figure 22: Software SPI functions diagram

ENC_functions

Handling functions of the ENC28J60 converter are defined in this library. They are based on the datasheet. A function table is shown in the Figure [23]. The functions can be divided into two groups. Functions fo operate the transmitter registers (encCtrlReg_Write, encCtrlReg_Read, encBit_set and encBit_clear) and functions for handling the buffer (encBuff_read, encBuff_read2, encBuff_write). Flowchart of these functions are in Figure [24].

Features encBit_set and encBit_clear are unique in that they do not work with the entire registry but only with individual bits. Function encBuf_read2 allows to read two bytes from the buffer at a time, which is used for instance in reading a position of the next packet or a vector packet status, which are 16-bit numbers. As the last of the primary functions of

this library is the function COMMUNICATOR_INIT that when you start or restart makes the basic setting of the transmitter and prepares it to receive messages.

Instruction Name and Mnemonic	Byte 0								Byte 1 and Following							
	Opcode			Argument					Data							
Read Control Register (RCR)	0	0	0	a	a	a	a	a	N/A							
Read Buffer Memory (RBM)	0	0	1	1	1	0	1	0	N/A							
Write Control Register (WCR)	0	1	0	a	a	a	a	a	d	d	d	d	d	d	d	d
Write Buffer Memory (WBM)	0	1	1	1	1	0	1	0	d	d	d	d	d	d	d	d
Bit Field Set (BFS)	1	0	0	a	a	a	a	a	d	d	d	d	d	d	d	d
Bit Field Clear (BFC)	1	0	1	a	a	a	a	a	d	d	d	d	d	d	d	d
System Reset Command (Soft Reset) (SRC)	1	1	1	1	1	1	1	1	N/A							

Legend: a = control register address, d = data payload.

Figure 23:ENC28J60 command protocol, source literature PDF [5]

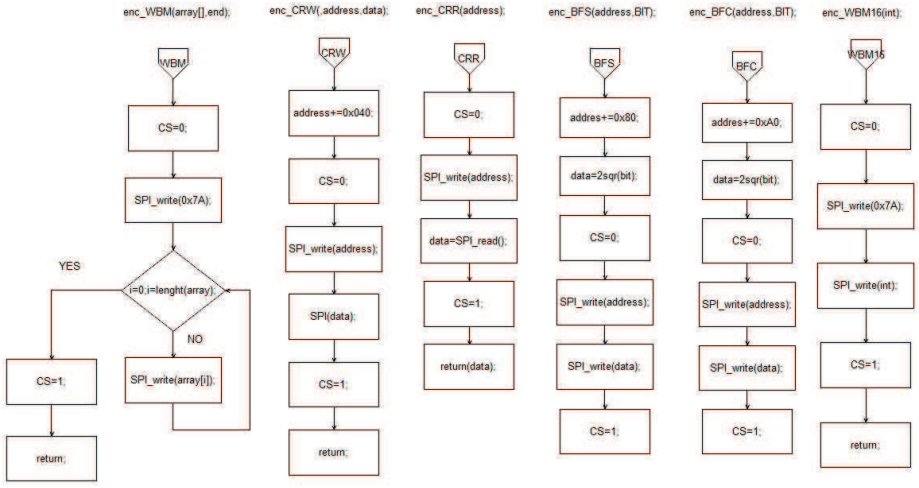


Figure 24: ENC28J60 commands flow charts

ETHERNET_functions

These functions are designed for processing the received packets. The basic function is read_packet_header that interrupts the service routine of INT0.

Function read_packet_header

Used to read the header of the received packet. The function uses the functions of the lower sheet ENC_functions. It reads and transmits data using global variables function evaluation_of_packet.

Function evaluation_of_packet

This function evaluates the header of the received packet, based on the type of the MAC address and STATUS vector packet. According to the evaluation function calls functions for processing the packet.

Function read_unicast_packet

If this is the unicast packet, a packet destined to one consignee, verified data from the header and raw data contained in the packet is stored in the memory on the I2C bus and the module status is changed.

Function read_reply_broadcast_packet

This function handles the broadcast packets. These packets are used to identify devices connected to the network. The function verifies the data contained in the packet and prepares a response packet and sends it to the recipient by the head received by the broadcast packet.

Function UDP_header_completation

In order to assemble the packet according to the standards so as not to be dropped by the active elements of the network or the receiving computer, you must first complete the protocol header in the descending order from the highest. The function is used to UDP_header_completation composition to create a UDP header as shown in the Figure [25]. CRC in the UDP header is not counted because it is not mandatory and does not affect the packet delivery.

Function IP_header_completation

This function completes the IP header. Unlike the header of the UDP protocol it is necessary to calculate the CRC which is checked when the packet passes through the active elements of the network. Packets with invalid CRC are discarded. Function diagram is in the Figure [25].

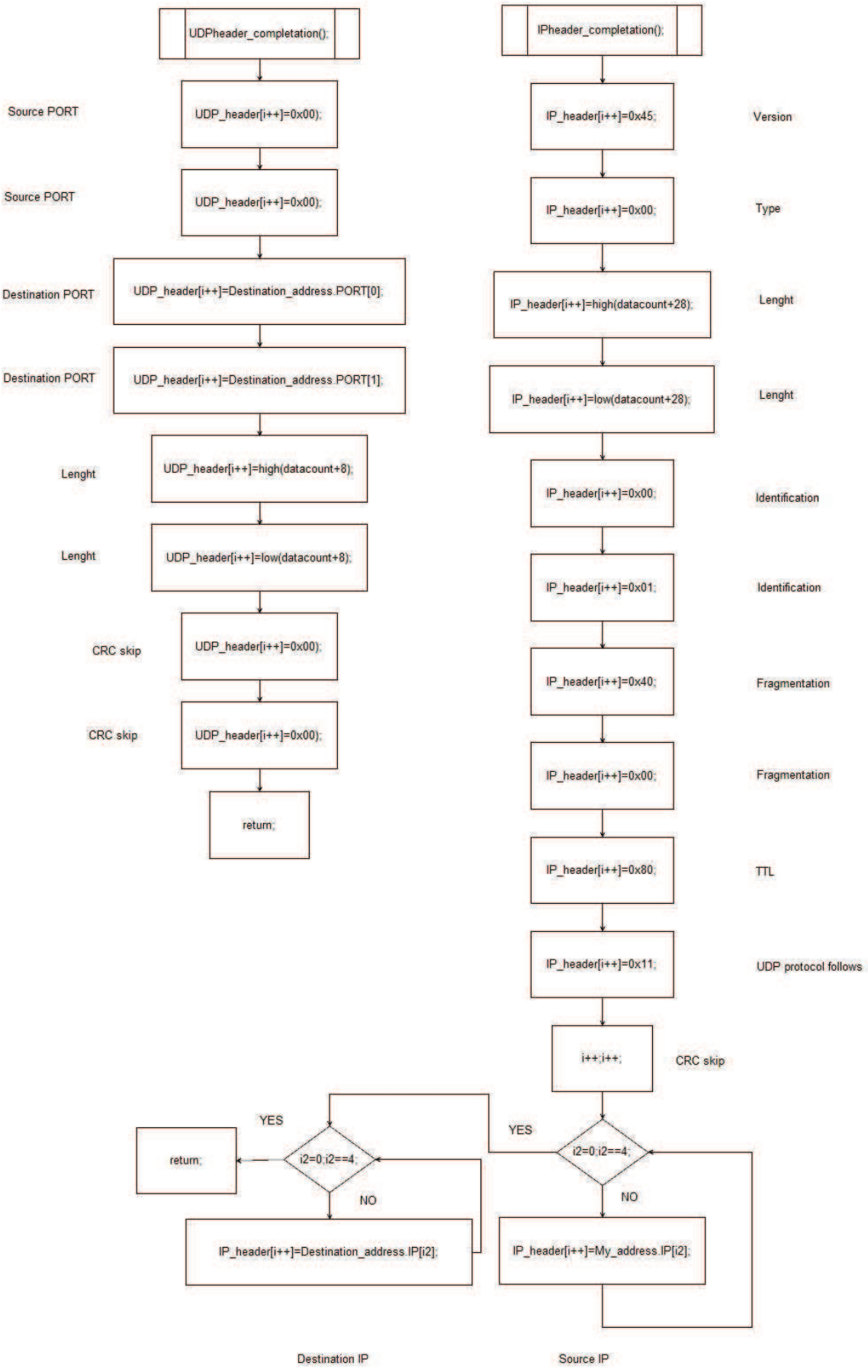


Figure 25: UDP and IP header completation flow chart diagram

The rest of service functions that are contained in the module are only useful functions for working with the data and their description is in the comments.

3.4. Software for PC

For the purpose of this thesis simple software for the communication with the Ethernet modules was created. When creating the software the decision had to be made between using Java and. C #. For the final selection of the programming language C # was decided on the grounds that the new OS Windows 7 and Windows 8 already contain .NET Framework and is not to be installed. Another reason was better compatibility with the operating systems from Microsoft and suitability for creating portable applications. Program created in C# is used to identify devices connected to an Ethernet network, reading the error message, identifying the actual value of the measurement cards or control elements, changing the address of the module and sending its own custom commands.

Because the application is intended to verify the functionality of the module only and maybe for any initial diagnosis, there are no additional functions available. However, other functions can be created using custom commands. After an implantation of the communication modules to the measuring cards and creating a system of identifiers of the elements will the application be extended by other types of commands for the each type of the device. For instance, the use of the module and the host computer of the tram brake Figure [26] by which could be a static function test The lever / release the brake implemented. The program is included on the CD. The main application window with a description of the main features is shown in the Figure [27].

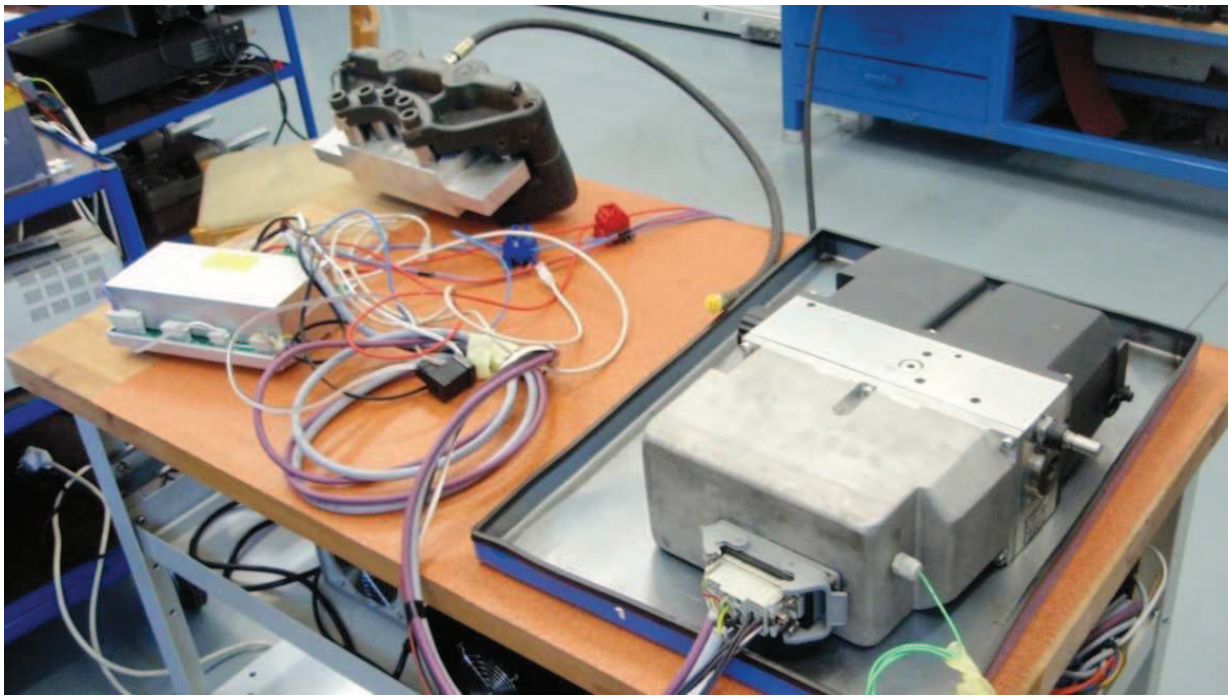


Figure 26: Static function test of the tram brake

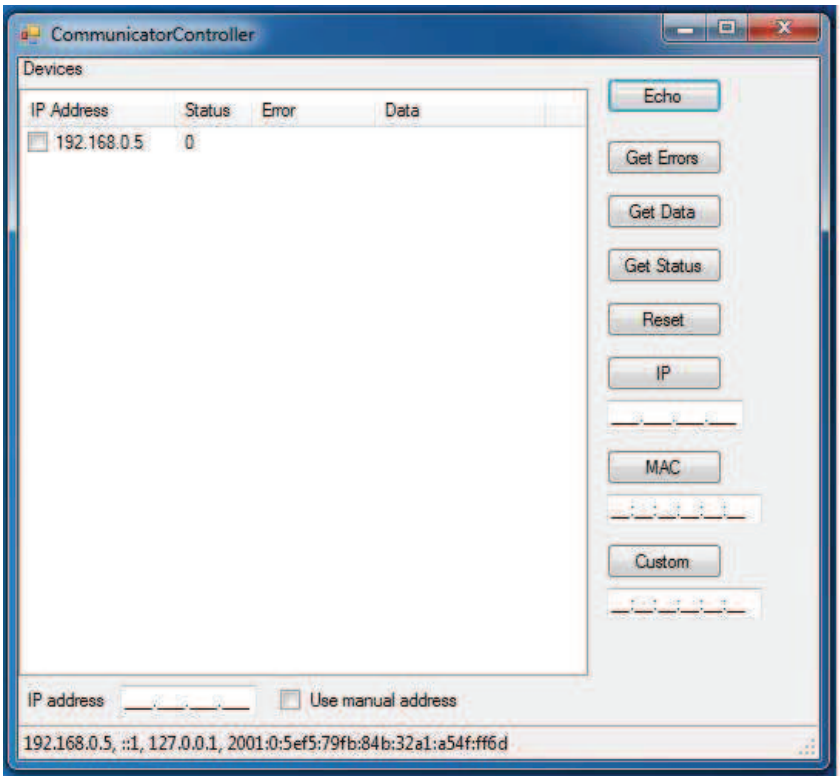


Figure 27: PC software main window

After starting the program it initializes the configuration file, it creates the main window, buttons and containers for the listing of the devices. Then it waits for the user input. Use the ECHO to send a UDP packet with the 0A0A as MAC address data set as a broadcast. After submitting the application, it waits for a response from the connected device.

According to the received unicast data packets with 0A0A application creates a list of connected devices with which it is possible to communicate. If the device is not found on the network, it is possible to try to connect with it manually by entering its address in the box at the bottom of the program. It is also possible to change the internal settings of the selected module using the IP and MAC addresses. RESET button resets the entire system including the master system. Buttons Get Errors, Get Status and Get Data are used for retrieving data from the host system. Data are saved as a text file as raw dataafter reading. Custom button allows you to send any command that is not yet implemented in the program.

3.5. Network trackers

For debugging the software module and an overview of the structure of the individual packets sent and received from the local network programs for monitoring the network were used. From the tested freeware programs the program Wireshark was selected. UDP packets which are used for the communication in the thesis have been captured with this program. It was also verified that the PC itself, or rather the operating system does not use UDP packets to attempt to identify the connected devices. In these experiments the operating system sends TCP and ARP packets, and waits for the response of the connected systems. It was necessary to have insight of these packets because of possible collisions when communicating with Ethernet modules, or to prevent congestion. Thanks to indetification of these packets it was possible to set both hardware ENC28J60 converter filters and software filters in ATmega MCU so the module would not accept such packets. Wireshark software also allows you to view data packets as a whole. It was necessary to identify the individual bytes packet in order to create custom packets in the MCU chapter „Ethernet data transmition“.

4. Testing

4.1. Operational tests

Functional verification has been performed by several tests on newly manufactured PESA 121N trams in Polish Warsaw and the tram EVO 2 of the Pragoimex company in Liberec. The first tests involved thermal stability of the system that has been tested in a climatic chamber. Since it was not possible to test a whole control card, an evaluation kit with the installed module was used. On the UTP cable connected to the transmitter was connected the RJ45 connector which had connected pins TX and RX. In order for it to be accepted when it does not properly reference the MAC and IP address, a Promiscuit mode was used. If this function is active, module receives any packets. Data contained in the packet was always incremented and displayed on the LCD. In these tests the temperature of -41 to +81 ° C has been achieved. Test in the climatic chamber is shown in the Figure [28].

Figure 28: Climatic chamber test

High temperatures did not show any signs of affecting the module. An error of ATmega MCU internal oscillator was detected at low temperatures. The disorder was manifested by reducing the frequency of oscillation from 8 MHz to 7.19 MHz, which is about 10%. This value is already significant, however, the function of the module was not significantly effected and SPI communication interface did not show any errors. Another interesting fact was that the crystal used for the converter ENC28J60 showed a much smaller error of the oscillations in the entire temperature range than the internal oscillator of the MCU.

In case of the necessity of the repairs of the internal oscillation of the MCU an external crystal could therefore also be used. In tests at temperatures below -35 ° C and above 80 ° C the LCD display had to be removed from the development kit on which the test was performed because of the risk of damage.

Further tests were focused on the functionality of the communication module with a PC during the operation. Given that the rail vehicle is powered by electricity, there are large changes of current flow, when there is a forward movement when the current is consumed or when braking when electricity is generated by an electromagnetic brake which leads to recovery. These changes and the overall closeness and connection of all devices lay great demands on shielding data cables and communication cables. This was verified using UTP (Unshielded Twisted Pair) cable for the Ethernet communication. The power of the interference to the signal cable can be seen in the Figure [29].

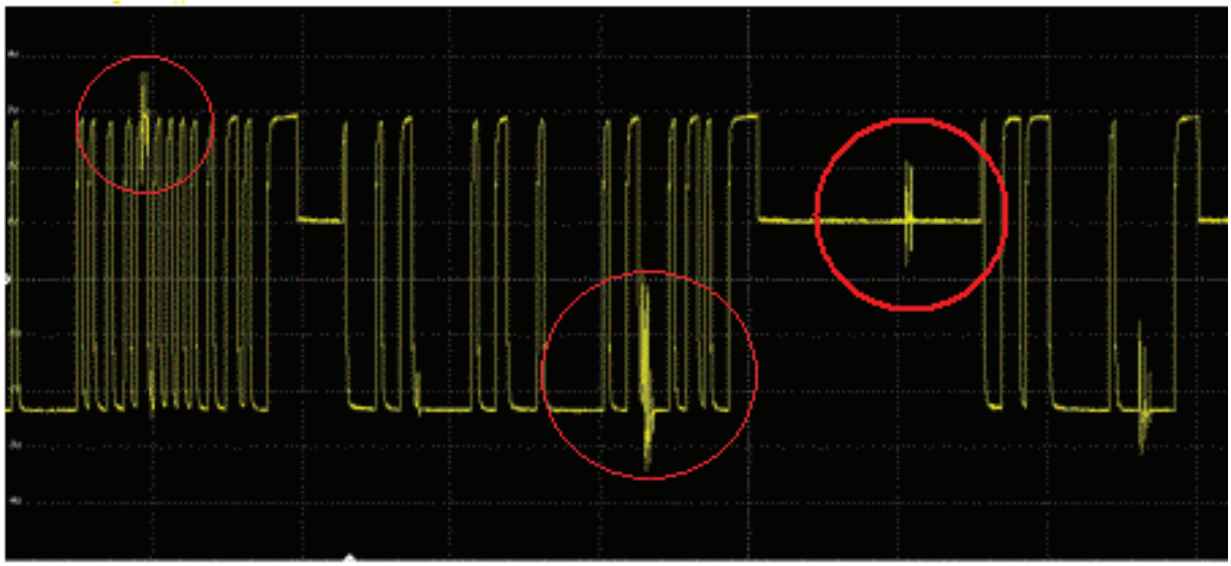


Figure 29: Electrical interference on UTP cable

For this reason, FTP cables (Foiled Twisted Pair) were used for the testing of the communication and communication modules located close to strong sources of interference were equippedby STP (shielded twisted pair)cables. Differences between the cables are shown in the Figure [30]. It is needed for an ethernet switch to be designed for the use of the STP.

Figure 30: The difference between the UTP and STP cable, source literature [26]

Retrieving data from a higher-level systems was verified using the PC After verifying the functionality of the communication modules from the chapter „Sotware for PC“. Since the module is primarily intended for reading diagnostic and error messages stored in the computer memory card, this verification has been made in the tram EVO-2 in Liberec which has already been put into operation to verify driveability. Thus error messages were already generated in the system, such as a power failure, low pressure in the brake aggregates etc.

Functional verification was successful, the program would be able to identify all the communication modules and make their identification and diagnostics. The problem was only with the module mounted on the control board of the drive motor, but later it turned out that a damaged PCB, which had an interrupted connection, was responsible for the malfunction.

4.2. Functionality and use

The communication module is fully tested both in the laboratory and under the conditions of testing it during driving the tram without major errors. The device is therefore fully functional SPI to the Ethernet converter, with the ability to communicate with PC by the UDP packets. Module operation host system is very simple and you can change the settings

for the module and run the program. The application fully complies with the requirements set by the company DI-ELCOM in a number of upcoming innovative systems for the rail vehicles. The construction of the communication module also meets economic requirements. Its price is about the same as in 10% of the cheapest available converters. Due to the number of the modules used on one rail vehicle is a financial saving significant.

Using the module in the tram PESA-121N, including the switch as a connection point located in the cabinet of main tram computer is in the Figure [31] and [32]. On this configuration were performed the test drives and functionality tests of modules.

Using this module, it is possible to connect any embeeded device with the SPI interface to an Ethernet network and use it to drive it by the communication module.

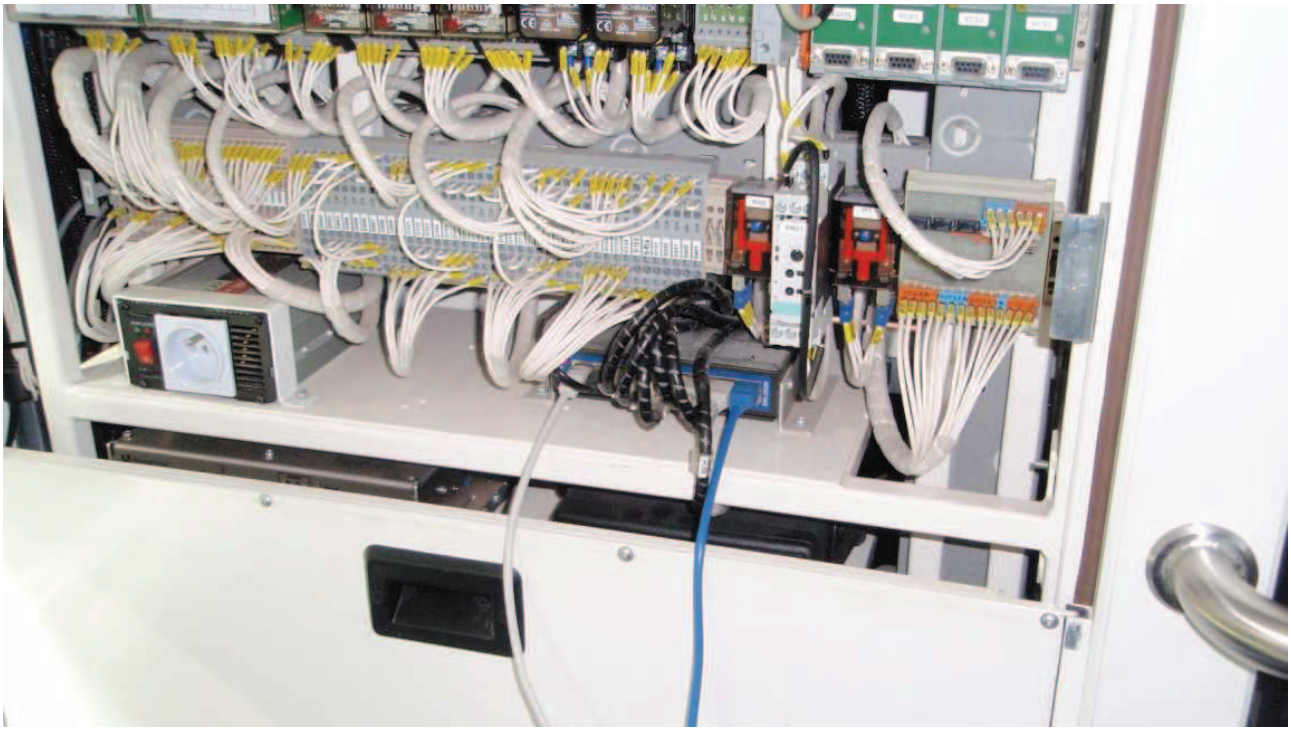


Figure 31: Detailed picture of the main tram computer with a industrial ethernet switch

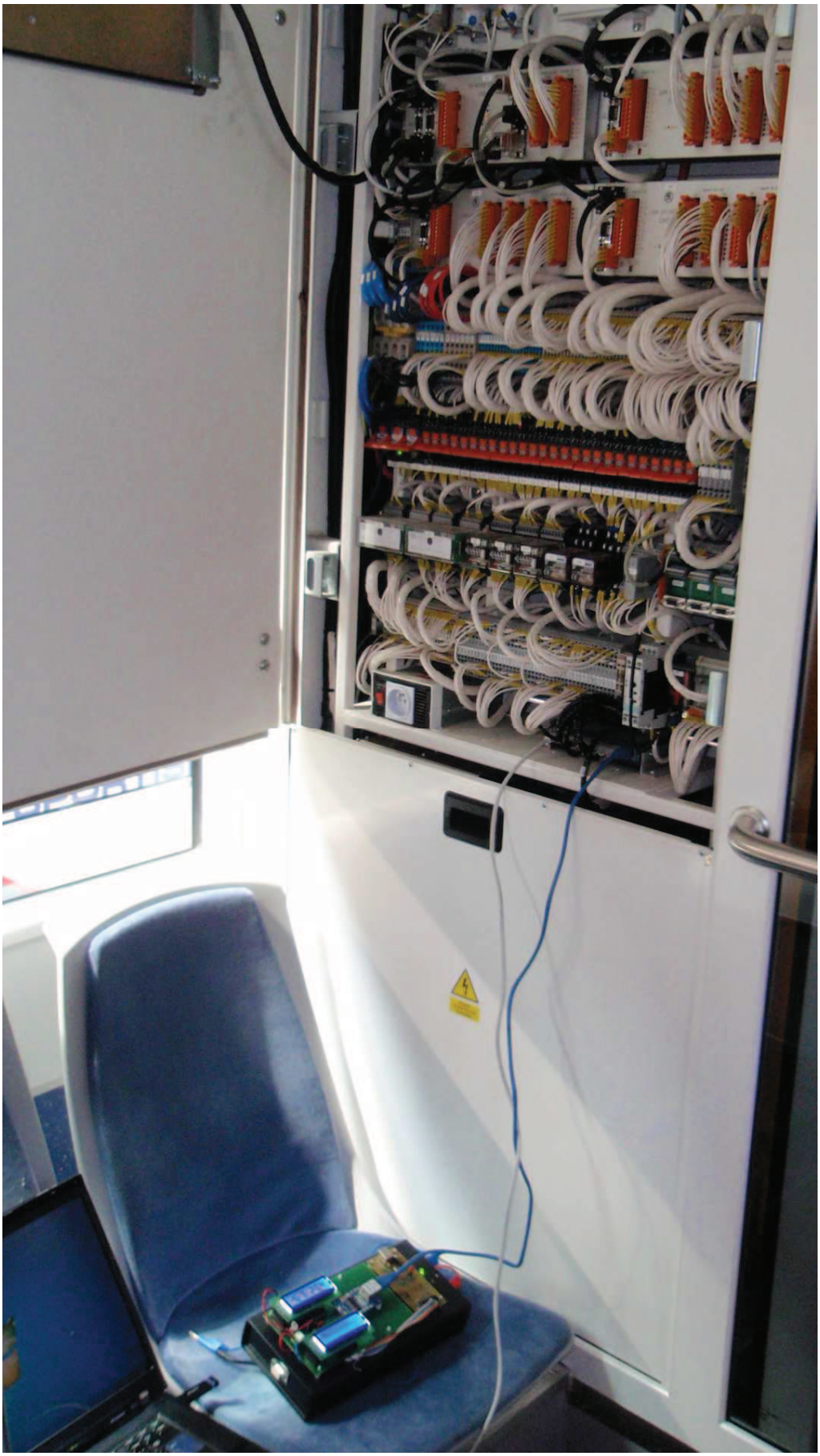


Figure 32: The main tram computer with connected development kit

5. Conclusion

Input idea for this thesis was the possibility of using an Ethernet interface for remote diagnostics of the centralized control and measurement systems using the Ethernet. Facilities possessing this interface can be easily attached to a central point, and allows remote administration and monitoring during the operation. Reading out data can be processed centrally regardless of the location of system elements. Another motivation for this work was the fact that similar devices do not commonly occur, or occur, but in most cases they are using other devices equipped with communication interfaces than it was required in the specification or are not affordable. After the basic analysis of the given topic the design and implementation of part of the facilities was carried out, under which it is possible to build a whole network of diagnostic points. This work deals with the selection of necessary components and their physical connections. Motion hardware part was made in form of development kit that allows the realization of the intended destination device, then the final module. This form of execution is also suitable for further editing of the software for use in other types of equipment. The basic components are the controlling MCU ATmega-8 and Ethernet ENC28J60 coverter completed by the RJ45 connector with the integrated magnetic circuits. Software is neatly divided into the libraries and using various interruptions, it is easy to update it for the new features and commands.

It is needed to finish the recording of the new firmware for the master system on the software module in the future. It did not happen during the thesis because of time. It would also be possible to prepare functions for the TCP communication protocols and to use the module for displaying data on the Internet, but this feature cannot be used for the control card for which the module was developed. Its versatility, however, allows connection of other devices which could usethis function. It would be also possible to use smaller the housing ENC28J60, namely housing SSOP-28 and make the overall miniaturization.

Software for PC is appropriate to modify and extend in the future. To invent the system of identifiers of the separate devices (control cards) function to offer the user commands for working with the selected system according to their functions. Given that the program for the PC was not the main focus of the thesis and not even a list of all managerial cards which should the program support was complete, adjustments were not implemented during the work.

The biggest problem with the diploma thesis was the communication via USART in the SPI mode with the ENC28J60 converter. Although it is stated in the datasheet, that it is possible to use communication in all four modes even after setting registers on the stated

values, the logical values of the SPI levels remained in the Mode 2, which corresponds to the original settings of the USART. This behavior is reflected in the three tested MCU ATmega-168. When asked about the MCU registry settings, manufacturer did not respond. According to the available information obtained from the other users of the internet forums this problem has not been solved anyone yet. I used the software SPI as a solution to the problem in the thesis. At the last optimalization of the program was foud that the problem with seting the USART-SPI of ATmega-168 to MODE 0 was in Codevision enviroment. The preprocesor of Codevision did not accept the C commands and did not load the given values to UCPOL and UCPHA registers. So the USART-SPI staied in the standart setting. The HP INFO TECH company, as a developer of the IDE was informed about that error, but in the diploma thesis was work with the software SPI as has been written.

According to written thesis it must be sad that developed modules meets the requirements for connecting embeeded systems to Ethernet. However, for manage the full access to the Internet would be necessary to implement ARP and TCP protocols. To implement these protocols would be appropriate to use a processor with 32bit architecture. This would allow better work with long numbers as the CRC, MAC, etc. These operations makes a heavy load in used MCUs ATmega-8 and ATmega-168.

Due to the fact that the scope of work was very wide, from the design and the manufacture of the hardware, through the programming MCU to create a "demo" version for the PC. I had to use large amount of knowledge acquired in the Bachelor's stage. I also have some knowledge of both widened and deepened. An important contribution was also cooperation with DI-ELCOM, and I gained knowledge and awareness of the cooperation development of complex devices and work in a team. Ultimately, for me, the work was rewarding and also very interesting.

Used literature

Books:

[1]. Vladimír Váňa. Atmel AV. BEN – technická literatura, 2003. 216 s. ISBN 80-7300-102-

PDF, contained on CD:

[2]. Atmel Corporation, ATmega8/L datasheet, 27.2.2013
[3]. Atmel Corporation, ATmega48/88/168 datasheet, 4.5.2011
[4]. Bell Fuse, MagJack datasheet, 20.8.1998
[5]. Microchip Technology, ENC28J60 Data Sheet, 19.11.2012
[6]. Dr. Jones, Rion Hollenbeck , The IEE 802.3 Standart (Ethernet), 17.9.2001
[7]. Marina del Rey, Information Sciences Institute, Internet Protocol, January 1980
[8]. J. Postel, User Datagram Protocol, 28.8.1980

Used internet links:

[9]. Ethernet examples 1 – hw.cz
http://www.hw.cz/teorie-a-praxe/konstrukce/ethernet-v-prikladech-i.html[6. 5. 2013]

[10]. Ethernet converters Gnome – papouch.cz
http://www.papouch.cz [6. 5. 2013]

[11]. Ethernet – wikipedia.org
http://cs.wikipedia.org/wiki/Ethernet[6. 5. 2013]

[12]. Ethernet crossover cable – wikipedia.org
http://en.wikipedia.org/wiki/Ethernet_crossover_cable[6. 5. 2013]

[13]. Ethernet – wikipedia.org
http://en.wikipedia.org/wiki/Ethernet[6. 5. 2013]

[14]. IEEE 802.3 – wikipedia.org
http://cs.wikipedia.org/wiki/IEEE_802.3[6. 5. 2013]

[15]. The IEEE 802.3 Standart (Etherner) – rionhollenbeck.com
http://rionhollenbeck.com/GradPortfolio/Papers/620-Ethernet/Ethernet.pdf[6. 5. 2013]

[16]. IP Protocol – zam.opf.slu.cz
http://zam.opf.slu.cz/botlik/CD-0x/5.html[6. 5. 2013]

[17]. UDP: User Datagram Protocol – pcvr.nl
http://www.pcvr.nl/tcpip/udp_user.htm[6. 5. 2013]

[18]. Network applications with microcontrollersAtmel – hw.cz
http://www.hw.cz/teorie-a-praxe/konstrukce/sitove-aplikace-s-mikrokontrolery-atmel.html[6. 5. 2013]

[19]. CodeVisionAVR (version 1.23.8b) – instruct1.cit.cornell.edu
https://instruct1.cit.cornell.edu/courses/ee476/codevisionC/cvavrman.pdf[6. 5. 2013]

[20]. MAGJack® - hw.cz
http://www.hw.cz/soucastky/magjackr-ihned-a-za-babku.html[6. 5. 2013]

[21]. Magjack datasheet – farnell.com
http://www.farnell.com/datasheets/34927.pdf[6. 5. 2013]

[22]. Differences between cables according to their designation
http://www.kabelazstrukturovana.cz/utp-ftp-nebo-stp-kabel[6. 5. 2013]

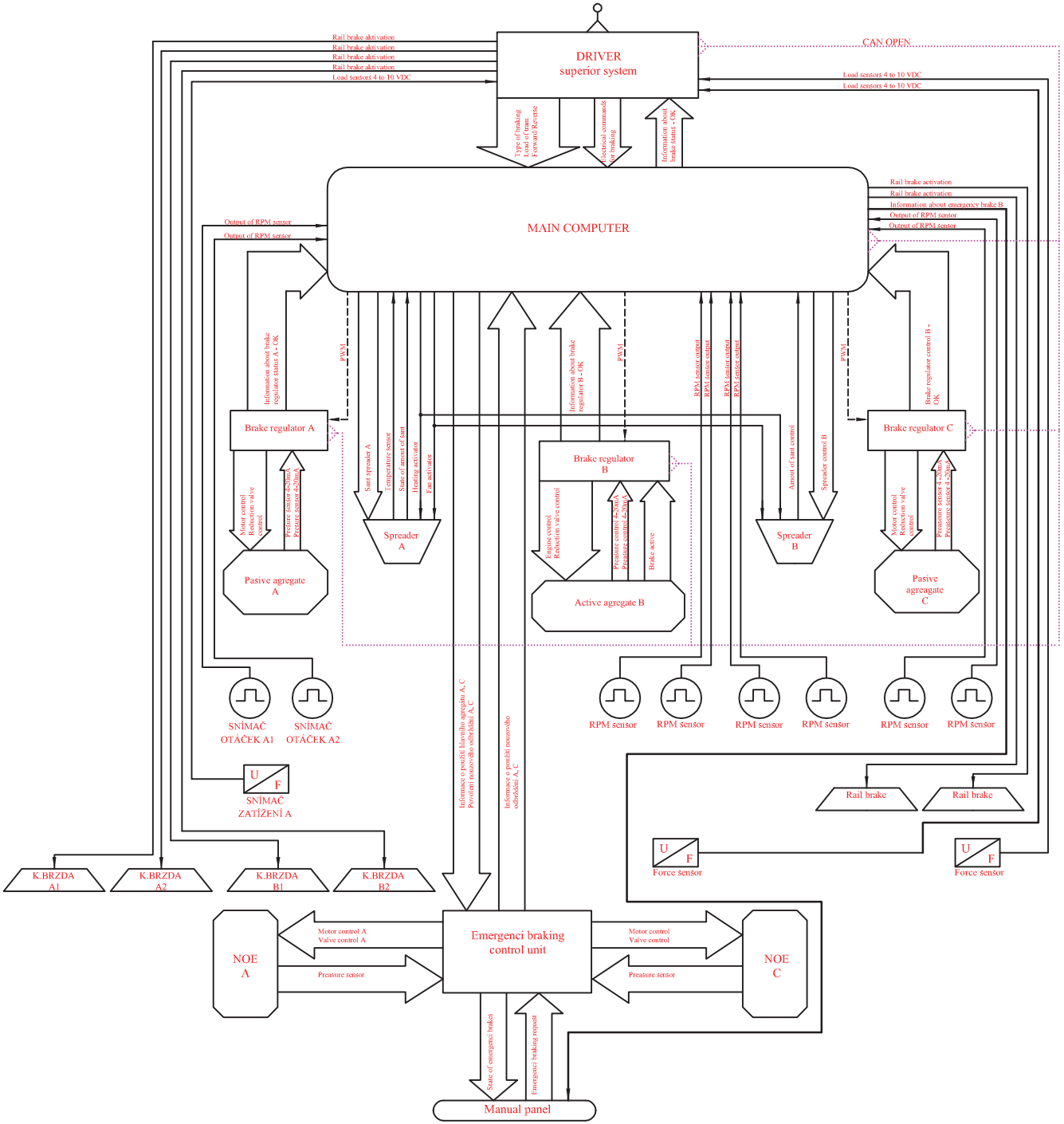
[23]. I²C vs SPI – byteparadigm.com
http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols [6. 5. 2013]

[24]. ISO JTAG-ISP programmer
http://sem.sch.bme.hu/projektek/avr_programozo_es_debugger[6. 5. 2013]

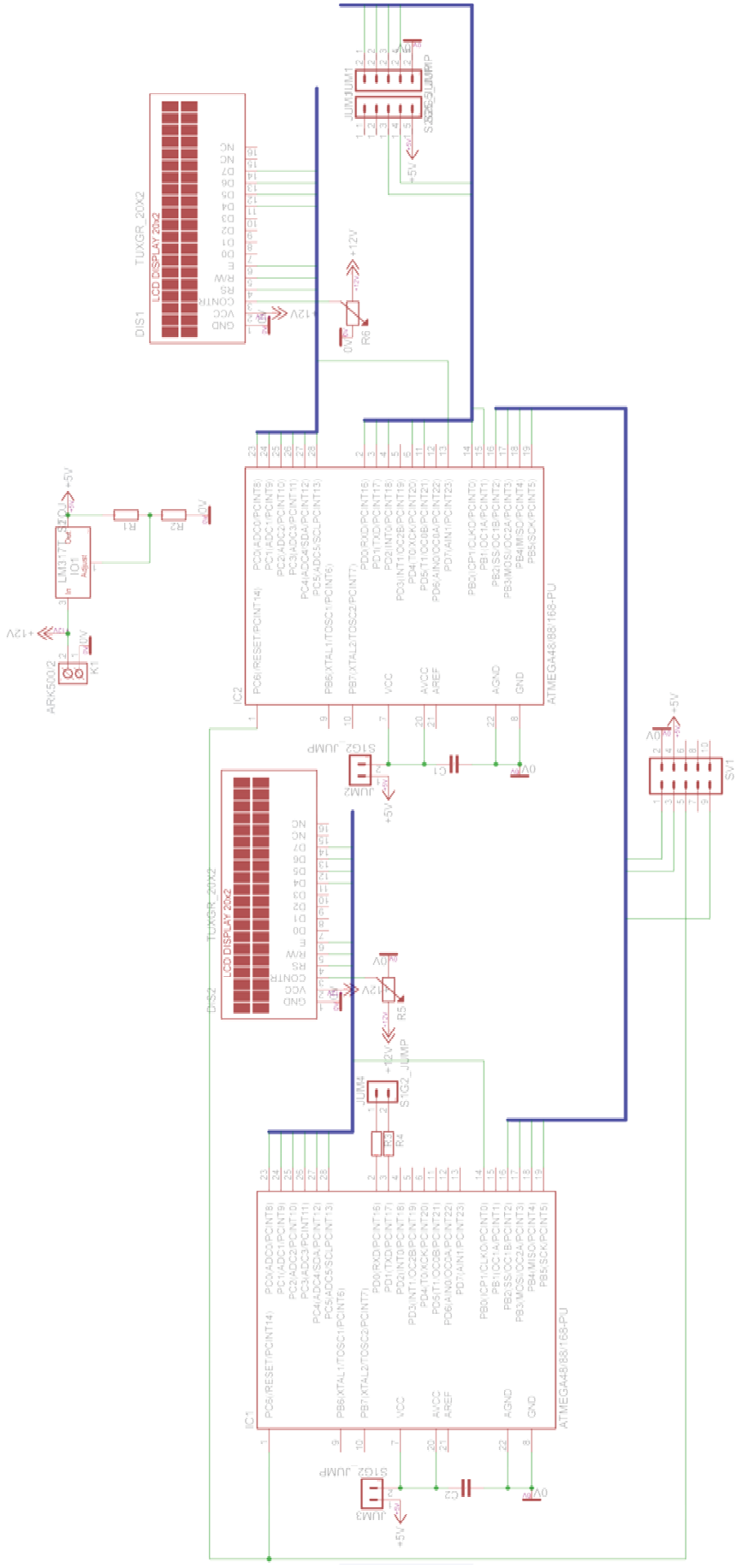
[25]. SPI modes
http://ez.analog.com/docs/DOC-2442[6. 5. 2013]

[26]. Ethernet Cables
http://multihow.net/2011/08/cat-5e6a-what-are-they-ethernet-cables-explained [6. 5. 2013]

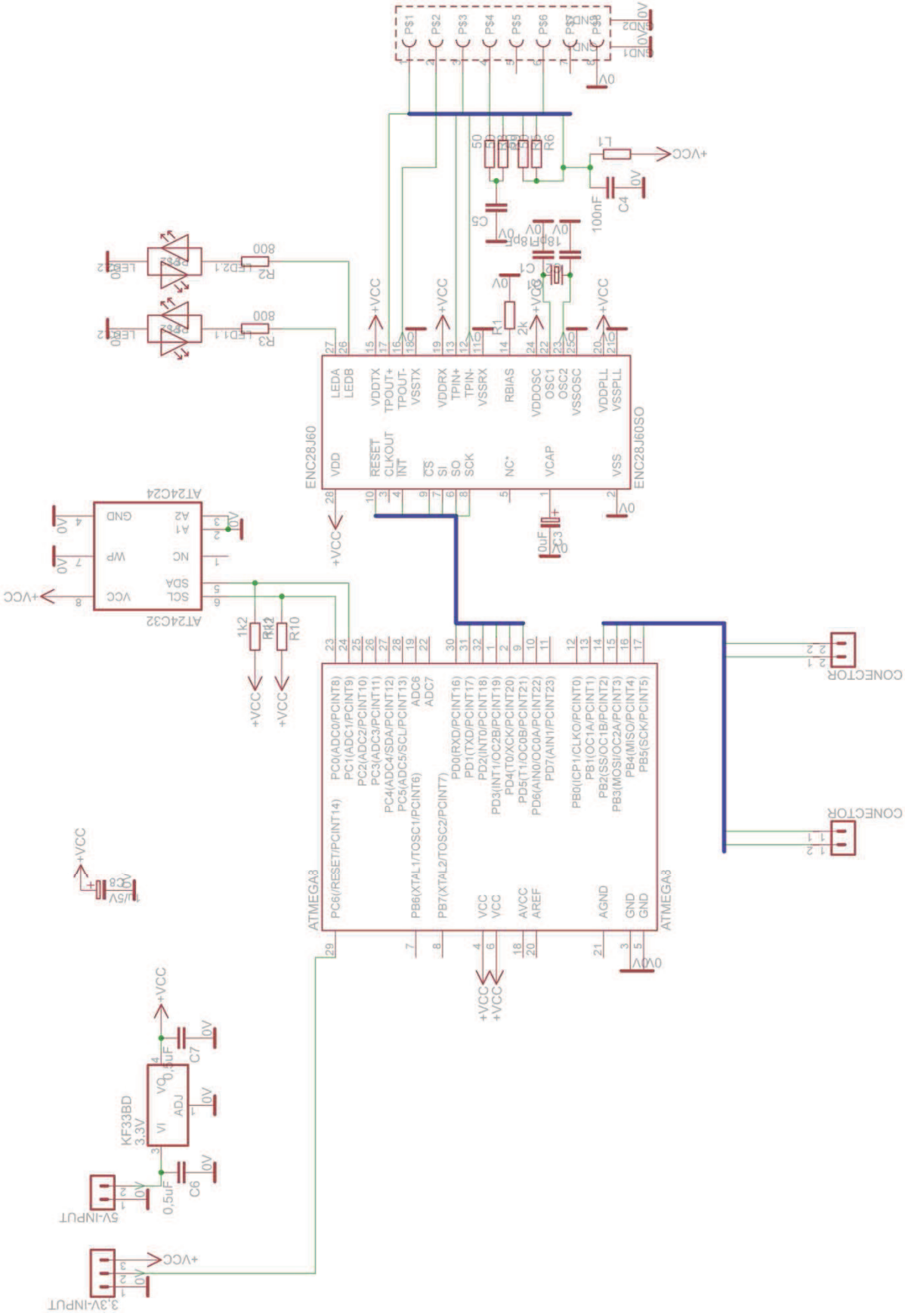
[27]. Ethernet aktive points
http://pctuning.tyden.cz/navody/zaklady-stavba-pc/7543-zaklady_pc-pocitacove_site_snadno_a_rychle?start=2[6. 5. 2013]



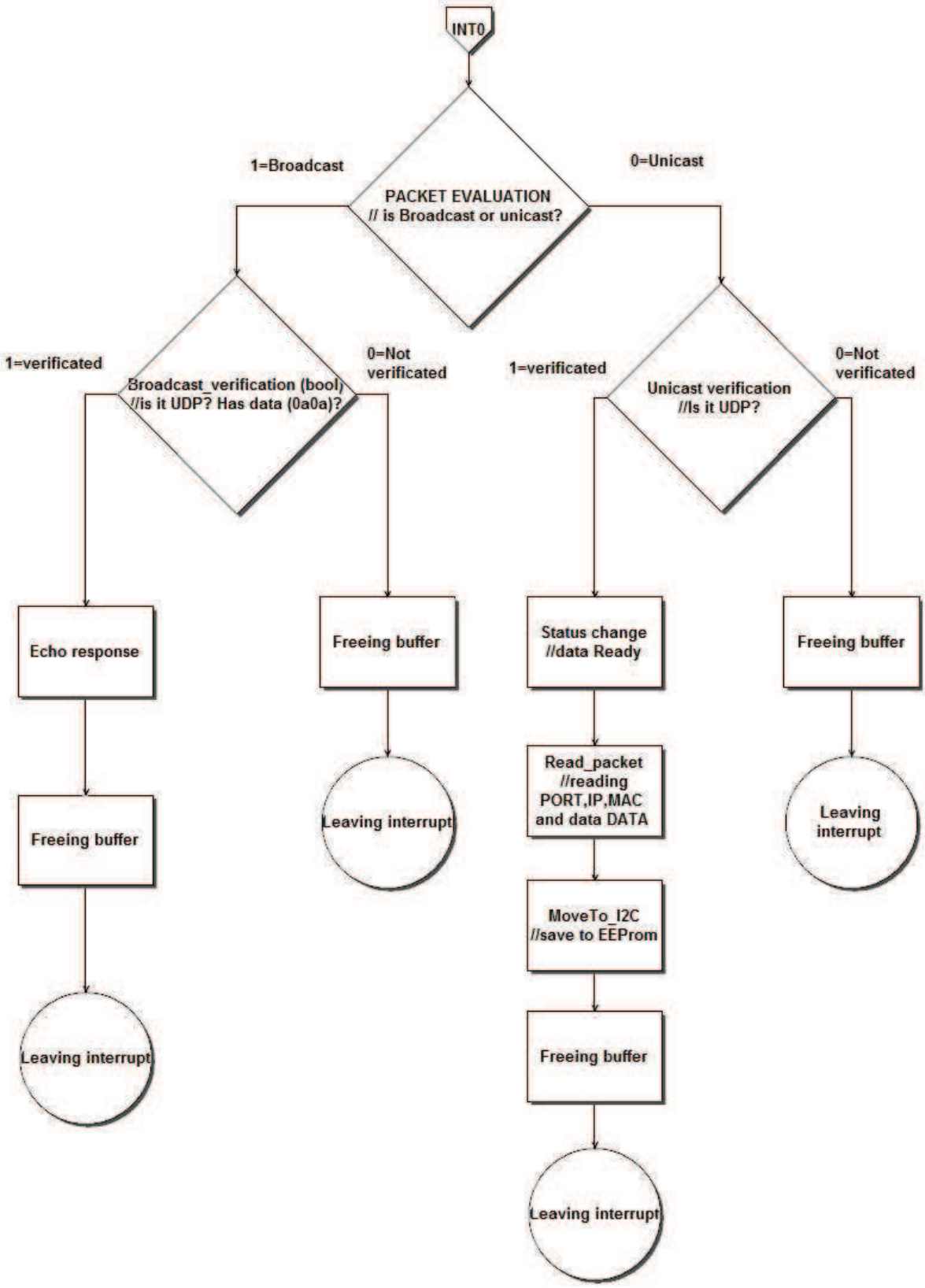
TOPOLOGICKÉ SCHÉMA TRAMVAJE 120N PESA WARSZAWA



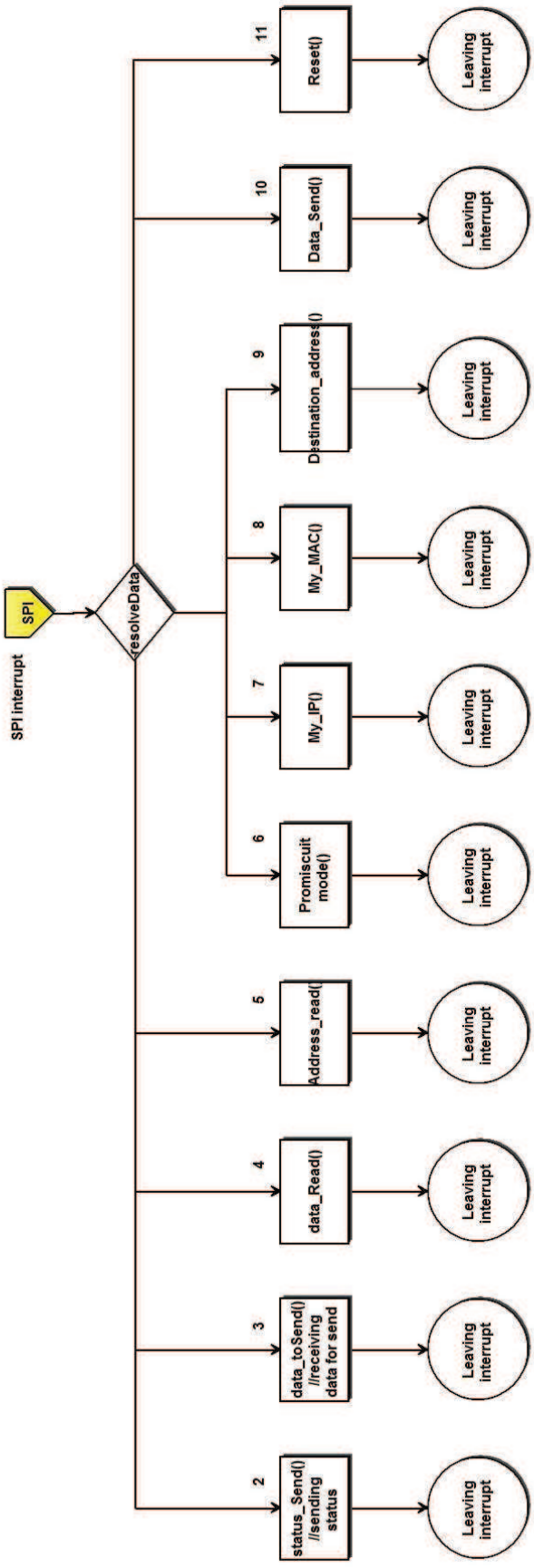
Annex 2: Wiring diagram of development kit



Annex 3: Wiring diagram of communication modul



Annex 4.1: Flow chart of INT0 interrupt



Annex 4.2: Basic flow chart of SPI interrupt