



TALLINNA TEHNIKAÜLIKOOL  
MEHAANIKATEADUSKOND



ITMO UNIVERSITY

Department of mechatronics

Chair of Mechatronic System Components

MHE70LT

*Valerii Altunin*

**Wireless Communication and Control of Autonomous  
Vehicles**  
Master's Thesis

Author applying for  
master's sciences of technical  
academic degrees

Tallinn  
2015

(The reverse of the title page)

## **AUTHOR'S DECLARATION**

I hereby declare that this thesis is the result of my independent work.

On the basis of materials not previously applied for an academic degree.

All materials used in the work of other authors are provided with corresponding references.

The work was completed ..... guidance

“.....” .....201...a.

The author

..... signature

The work meets the requirements for a master's work.

“.....” .....201...a.

Supervisor

..... signature

Permit to defense

..... curriculum defense superior

“.....” .....201... a.

..... signature

## MASTER'S THESIS SHEET OF TASK'S

Year 2015 semester 2

Student: Valerii Altunin 152334 MAHM02/13 (name, cod)

Curricula: Mechanics

Spetsiality: Mechatronics

Supervisor: Head of Chair, Professor Trieu Minh Vu (position, name)

### MASTER'S THESIS TOPIC:

(in English) Wireless Communication and Control of Autonomous Vehicles

(in Estonian) Autonoomsete sõidukite traadita ühendus ja kontroll .....

### Thesis tasks to be completed and the timetable:

Nr	Description of tasks	Timetable
1.	Finalize the thesis tasks. Writing the working algorithm.	02.03.2015
2.	Build the network and test data transmitting. Programming of microcontrollers.	09.03.2015
3.	Build the autonomous vehicle system. Writing article	13.03.2015
4.	Gather data from sensors. Their analyzing with MATLAB.	21.03.2015
5.	Complete the Thesis with all contents	15.05.2015

**Solved engineering and economic problems:** Hardware requirements for nodes construction, software requirements, development of smart-sensor wireless network, development of controlling algorithms and software, using of less energy consuming hardware, automatisations of vehicles.

### Additional comments and requirements:

.....

**Language:** English

Application is filed not later than 12.05.2015

**Deadline for submitting the theses** 22.05.2015

**Student** Valerii Altunin /signature/ ..... date 25.03.2015

**Supervisor** Trieu Minh Vu /signature/ ..... date 25.03.2015

## **Acknowledgment**

I would like to thank Professor Trieu Minh Vu, Professor Viktor Musalimov and PhD Stanislav Reznikov. Thank you for your help and valuable guidance and feedback. My special gratitude goes to Professor Mart Tamre, Engineer Svetlana Gromova and PhD student Sergey Zhigailov. I would also like to thank department of Mechatronics of ITMO University and department of Mechatronics of Tallinn University of Technology who provided me the opportunity to do mine science work and helped me in questions of training and organizational part. Finally, I would like to thank the ITMO University who financially provided my visit and study at the Tallinn University of Technology.

## Referaat

Antud töö on valminud Tallinna Tehnikaülikooli ja ITMO (Sankt-Peterburg) kombineeritud diplomiõppekava raames. Töö eesmärgiks on autonoomsele ratastega robotplatvormile juhtmevabal sidetehnoloogial põhinev juhtsüsteemi projekteerimine.

Igapäevases elus võetakse üha rohkem kasutusele juhtmevabal sidel põhinevaid tehnoloogiaid. Antud sidelahendusel on mitmeid eeliseid, mida saab rakendada tööstuses, näiteks: kaugjuhtimine ja monitoorimine; nutikad sensorvõrgud; automaatne võrku ühendamine. Seega, juhtmevabal ühendusel põhinevad juhtsüsteemid on antud tehnoloogiaarengu järgmine etapp ning on järk- järgult asendamas juhtmepõhiseid tehnoloogiaid.

Antud töös on käsitletud 802.15.4 standardil põhinevat mobiilse ratastega platvormi juhtsüsteemi. Juhtimine põhineb mitte ainult juhtkeskuse ning platvormi sidel vaid ka platvormide omavahelisel sidel.

Probleemi püstitus, olemasolevad lahendused ja näited tööstusest on toodud esimeses peatükis. Teine peatükk kirjeldab olemasolevaid juhtmevabu andmesidestandarddeid ning ratastega platvormi komponentide valikut. Kolmandas peatükis on käsitletud võrgusõlmede riist- ja tarkvara arendust, juhtseadme tarkvara arendust ning võrgu seadistamist. Kogu süsteemi funktsionaalsuse testitulemused on toodud neljandas peatükis ning viiendas peatükis tehakse järeldused ning kirjeldatakse edasist tegevust.

## **Abstract**

This work was done in frame of a double degree program between ITMO University and Tallinn University of Technology. Purpose of this project – is to develop a wireless network for controlling of autonomous wheeled platform.

Technologies of Wireless transmission are increasingly penetrated into everyday life. This type of communication has several advantages, which can be applied in solving of production problems: the possibility of remote control and monitoring, smart sensor networks, automatic connection to the network. Thus, the control of technical devices based on wireless communication is the next step in the development of control methods and is gradually replaced the control systems which use wires.

Using of wireless networks based on 802.15.4 standard for control of mobile wheeled platform considered in this article. Control is carried out not only from the control center, but also by the platforms, through their communication with each other.

The first chapter will discuss state of problem, existing solutions and industrial examples. Chapter two describes the existing network standards and approaches to the choice of a wheeled platform components. In chapter three considered hardware and software development of nodes of network, development of software for control center, and configuration of the network. Chapter four describes the experiments to test the functionality of the entire system and in the fifth chapter of general conclusions and future scope.

## Реферат

Данная работа была проделана в рамках программы двойного диплома при поддержке ИТМО университета и Таллиннского Технического университета. Цель работы – разработать беспроводную сеть для управления автономными колесными платформами.

Беспроводные способы передачи данных все больше проникают в повседневную жизнь. Такой тип коммуникации имеет ряд достоинств, которые могут быть применены в решении производственных задач: возможность удаленного управления и наблюдения, умные сенсорные сети, автоматическое подключение к сети. Таким образом, управление техническими устройствами на основе беспроводного взаимодействия является следующим шагом в развитии методов управления и постепенно заменяет управление с использованием проводов.

В данной работе рассматривается использование беспроводных сетей на базе стандарта 802.15.4 для управления подвижными колесными платформами. При этом управление осуществляется не только из контрольного центра, но так же самими платформами, за счет их общения друг с другом.

В первой главе будет рассмотрено состояние вопроса, имеющиеся аналоги сетей и промышленные примеры их использования. Глава два содержит описание имеющихся стандартов сетей и подходы к выбору компонентов одной колесной платформы. В главе три разбирается аппаратная и программная часть разработки узлов сети, разработка программного обеспечения центра управления, настройка параметров сети. В главе четыре описываются эксперименты для проверки работоспособности всей системы, а в пятой главе сделаны общие выводы и возможности для работы в будущем.

## Content

Introduction .....	1
1. Literature review .....	4
2. Methodology .....	9
2.1 Wireless technology.....	9
2.1.1 Bluetooth.....	9
2.1.2 Wi-Fi.....	10
2.1.3 IEEE 802.15.4.....	11
2.2 Microcontrollers .....	13
2.3 Sensors.....	18
2.3.1 Infra-red sensors.....	19
2.3.2 Ultrasonic distance measurer .....	21
2.4 Conclusion .....	22
3. System design .....	24
3.1 Hardware design .....	24
3.1.1 Design of node .....	24
3.1.2 Design of control center.....	31
3.1.3 Conclusion .....	32
3.2 Deployment of network .....	34
3.2.1 Requires and configuration.....	34
3.2.2 Testing the network.....	37
3.2.3 Conclusion .....	38
3.3 Depending function .....	39
3.4 Software design .....	43
3.4.1 Software for LPC H2148 .....	43
3.4.2 Monitoring and control .....	49



3.4.3	Conclusion .....	52
4.	Results .....	53
4.1	Autonomous behavior.....	53
4.2	Testing of full system .....	53
4.2.1	Testing of communication .....	53
4.2.2	Testing PWM output.....	55
4.2.3	Reaction of motors .....	56
4.3	Testing of GUI.....	57
4.4	Conclusion .....	58
5.	Discussion .....	59
5.1	Conclusion .....	59
5.2	Future scope.....	60
6.	References .....	62
	Appendix .....	64

## List of abbreviation

<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>WPAN</b>	Wireless personal area network
<b>FHSS</b>	frequency hopping spread spectrum
<b>RAM</b>	Random access memory
<b>ARM</b>	Advanced RISC Machine
<b>AVR</b>	Advanced Virtual RISC
<b>IDE</b>	Integrated development environment
<b>LED</b>	Light-emitting diode
<b>CCD</b>	Charge-Coupled Device
<b>MCU</b>	Microcontroller Unit
<b>GUI</b>	Graphical User interface
<b>UART</b>	Universal Asynchronous Receiver-Transmitter
<b>I/O</b>	Input / Output
<b>IR</b>	Infra-red
<b>PWM</b>	Pulse-Width modulation
<b>ADC</b>	Analog-Digital converter

## **Introduction**

Control, from the philosophy point of view, it is activities of changing object to achieve a certain goal [1]. In daily life the control could be found everywhere: driving of car, piloting of airplane, manipulation with industrial robots, opening of simple apartment door, switching-on of light. Depending on the complexity of the control system, it is necessary to use different approaches: use of mechanical switches, close the circuit, dropping the signal by wire or broadcast radio frequency with the instructions for action. Thus, the control methods can be divided into mechanical, electrical-wire and electrical-wireless.

### **Mechanical control**

With the advent of the first mechanisms was used mechanical control. In most cases, mechanical control systems include body of manipulation (steering wheel, lever etc.), mechanical executive channel (cable, shaft, etc.), the controlled object (blade, wheels, etc.).

Advantages of such control system:

- low price of one node
- ease of placement in the facility management

Disadvantages include the following:

- increasing of device functionality causes complexity in control system
- restrictions on use for complex devices
- a large mass of final system

With the development of technique for eliminating disadvantages began to use wire type of connection for controlling the objects

### **Drive-by-wire**

Wire control system helped to annul heavy mechanical transmission elements and replace them with wires. Thus, control of devices is carried out by supplying of control signal from the control center to the actuators.

### The Advantages

- Replaces mechanical elements to achieve the best values of weight and size.
- Simplifies maintenance.
- There is an opportunity to enter complex control algorithms that are physically impossible for human.
- System smarter and faster, and fully controlled by operator.

### Disadvantages

- The high cost of equipment
- Difficulties in ensuring reliability.

### **Wireless control**

The next stage in development of remote controlling it is wireless controlling. It is a logical extension of the wired control where physical components are replaced with radio waves. Thus do not need to think of cabling options, do not deal with the selection of the material from which wires are to be made, reducing the cost of the entire control system. There are only two elements: the control center and the end device.

### The advantages include

- Speed settings and scan the network manipulated
- Low final cost

### Disadvantages

- Lower bit rate signal
- Noises
- Programming complexity

At present in the field of automation widespread idea about development a systems, which can be controlled remote without using a lot of wires or large mechanism. Now this idea more and more realized in designing of “smart” home systems and automobiles [2]. There some parts replaced by nodes with autonomous energy supply and radio elements to communicate with control center.

Based on terms node in wireless network should work long time on autonomous energy source energy consumption of components of this node should be as less as it possible. For this purpose in field of development of microprocessor technology and radio technology were designed a special microcontroller units and communication standards.

One of those radio standards is IEEE 802.15.4

Objective of this work – to develop a wireless network, based on the IEEE802.15.4 standard, of wheeled autonomous platforms, with the ability to manage their movement from the control center, and in case of absence of control center, the possibility of communicating with each other and making decisions about their movement. Control center should provide a possibility to start or stop the movement of vehicle and let the user track the state of system. Moreover, control center should be presented as software with friendly graphical user interface (GUI).

## 1. Literature review

802.15.4 protocol is a relatively new protocol for transferring data wirelessly. The first mention about the possibility of its use in the fields of automation of the electronic equipment dates back to 2006 [3].

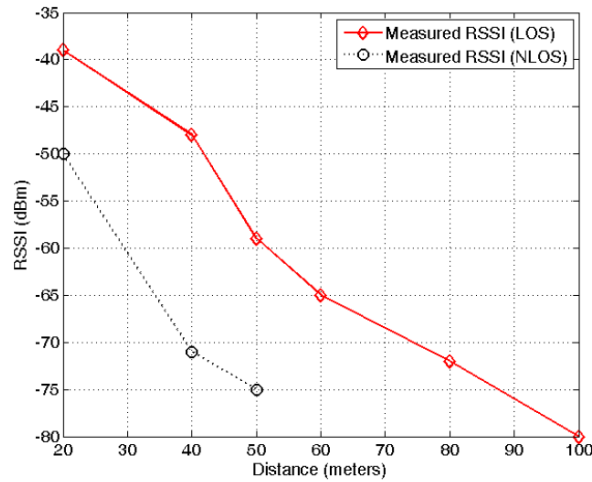
Wireless communications specification IEEE 802.15.4 has been present on the market since the advent of wireless data transmission BlueTooth and Wi-Fi. The emergence of that specification is primarily due to the fact that for some applications the main criteria for choosing a wireless transmission technology is the low power consumption of the hardware and its low cost. It follows a small bandwidth, as in most cases the power sensor is provided by built-in battery operating time from which exceed a few months and even years. Otherwise, a monthly replacement of battery radically changes the attitude of the user to the wireless technology. Existing at that time wireless data technology BlueTooth and Wi-Fi did not meet these criteria, delivering data at high speeds, with high levels of energy consumption and the cost of the hardware. In 2001, a working group number 4 IEEE 802.15, start to work on a new standard that would meet the following requirements:

- very low power consumption of hardware that implements the technology of wireless data transfer (battery life should be from a few months to several years);
- transmission of information should be carried out on non-high speed;
- low-cost hardware.

The result was the development of the standard IEEE 802.15.4. Zigbee Technology - further development based on 802.15.4 specification, which also determine access to the applications, the level of network and security.

In [4] researches attempt to create a one of the first network with using XBee. Simple network including three nodes: one coordinator and two end device. In this paper author considers, mostly, hardware part of project.

Paper [5] aimed to research work distance of XBee radio modules and its depending from number of nodes. There were evaluated received signal strength and delay. Evaluation was carried out for two cases: when in test space were some obstacles and when in test space was nothing (fig.1).



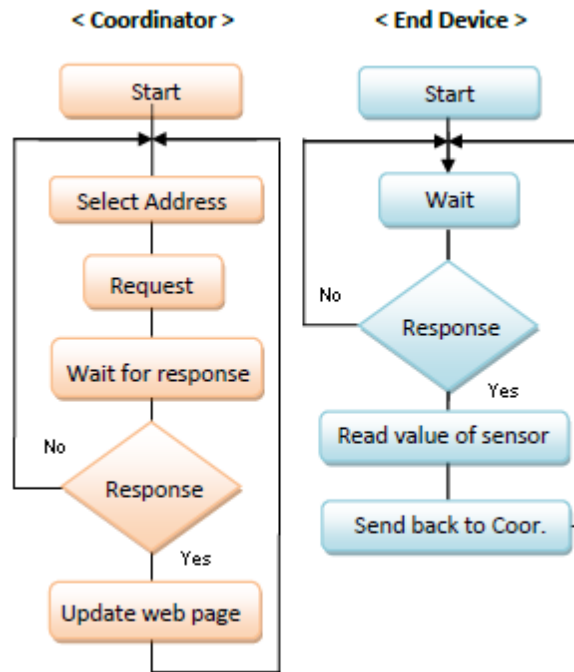
**Figure 1. Measured RSSI values depend on distance in LOS and NLOS**

During experiments were studied the data about the state of the signal of end node which was placed on some distance from the coordinator. As expected, if the distance increases the quality of received signal decreased. Thus, adding of obstacles, such as walls, will cause disconnect from the network on smaller distances. Based on those values it is possible to understand on which distances net can work.

One such example is the use of the wireless technology in project of WISE Museum [6]. The main focus in this article is on the location of the sensor nodes in the museum for optimal tracking state of environment. Also, a comparison with similar systems was carried out. Those systems based on more expensive equipment, or a larger components, or with use a lot of wires. Authors consider that, in the project of WISE Museum one of the problems of nodes - limited supply resources. However, given the scantiness of supply needed for ZigBee radio module, the problem is not so important.

System for controlling of temperature via internet was presented in paper [7]. There nodes with temperature sensors are placed in different rooms. A user requests via the Internet value of the temperature in a particular room, and the node activate the sensor, measure the temperature and sends the value back to the user.

Flowchart of Coordinator and End Device present on fig.2.



**Figure 2. Flowchart of Coordinator and End Device**

Same project was presented in [8] where some examples to which elements of home ZigBee technology can be applied. Also controlling of home environment via Internet is considered.

The same technology has been used in the field of security houses [9].

Advantages of ZigBee find their appliance not only in smart home, but also in air pollution monitoring systems [10]. This system provides to monitor the state of air as remotely via Internet as on the place where node located.

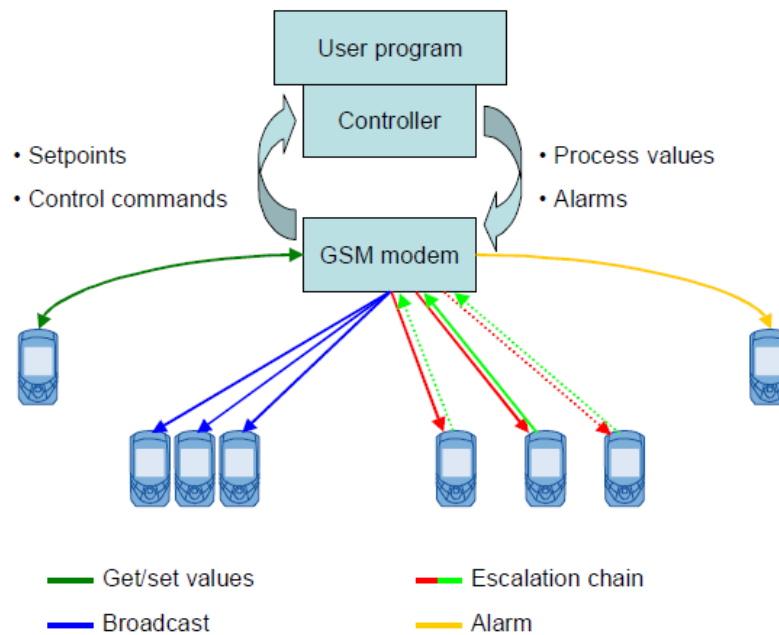
Using wireless communication system based on ZigBee in agriculture [11] provide highly economically efficient because it require less human power, increase numbers of controllable devices and reduce energy costs.

IEEE 802.15.4 is also suitable for medicine sphere. It provides to doctor actual information of status of patients even if doctor far away from hospital room [12].

XBee radio modules can be used for engines control. In one case Xbee is used as a receiver to switch states of servo-motors of the system [12], and in the other as a receiver for controlling by all parameters of a DC motor [13].



It is known using of technology to transfer data between end devices and the coordinator in commercial sphere. The idea of a wireless transmission device status has been raised in a scenario [15] where the data transmit over GSM. The technical solution proposes to monitor the state of the drink vending machine via GPRS. For this purpose the machine has a SIM card inside GPRS module which placed in machine (fig.3). The information is sent to the mobile phones users and administrators. For controlling of machine is also used the mobile devices. Managers have a specific numbers and sending data to the machine by SMS messages. The main difficulty lies in the "sorting" of information. There is a view that subscribed users do not have to get any "alarming" information. Such signals should be sent only to the phones of security services and supports. And also if broadcast users try to send "control value" it should not be recognized like a control signal.



**Figure 3. Model for information exchanging**

A Smart environment project was developed by Software Engineering Competence Center. In the technical report [16] the technical details of the infrastructure including the set of packets communicated between the nodes, the hardware and software design of the individual nodes to monitor the environment state were presented.

In [17] consider task of increasing number of nodes in network.

Thus, can be concluded for using of a particular technology should take into account both – its advantages and disadvantages. It is also worth to clearly understand how fast

data transfer should be done, how critical energy consumption of transmitters, the level of safety required, requirements to the integrity of the transmitted data, the number of devices in the network, etc. [18]. In the reviewed articles and papers presented results of practical experiments with devices that are based on 802.15.4 technology and commercial uses of this technology.

It should be noted, most of the relevant articles means the use of static nodes containing XBee modules. In the current work to make mobile node is supposed.

## 2. Methodology

To reach the objective formulated in introduction part there are four tasks should be solved:

1. Assemble an autonomous wheel platform
2. Deploy a wireless network based on 802.15.4 specification
3. Apply wireless network to control wheel platforms with LPC2148 development board
4. Develop a monitoring center with control and monitoring program in MATLAB

### 2.1 Wireless technology

#### 2.1.1 Bluetooth

Bluetooth - manufacturing specification of wireless personal area networks. Bluetooth allows the exchange of information between devices such as personal computers (desktop, handheld, laptop computers), mobile phones, printers, digital cameras, mice, keyboards, joysticks, headphones and headsets for reliable, free, widely available radio frequencies for short-range communication. Bluetooth allows these devices to communicate when they are within 100 meters (see table 1) of each other (range depends strongly on the obstacles and interference), even in different rooms.

The principle of operation is based on the use of radio waves. Bluetooth radio communication is carried out at ISM-band Industry, Science and Medicine, which is used in various consumer devices and wireless networks (free from licensing 2,4-2,4835 GHz range) [19]. In the method of Bluetooth frequency hopping spread spectrum - FHSS. FHSS method is simple to implement, provides stability to broadband noise, and inexpensive equipment.

<b>Class</b>	<b>Max.power, dBm</b>	<b>Work radius, m</b>
<b>1</b>	20	100
<b>2</b>	4	10
<b>3</b>	0	1

**Table 1. Power of signal and max. range**

The algorithm FHSS, the carrier frequency in the Bluetooth signal abruptly changes 1600 times per second (total released 79 operating frequencies of 1 MHz in width, and in Japan, Spain, France and the band has - 23 frequency channels). The sequence of switching between frequencies for each compound is a pseudo-random and known only to the transmitter and receiver, which every 625 microseconds (one timeslot) synchronously rebuilt single-carrier frequency to another. Thus, if the adjacent pairs are several transmitter-receivers, then they do not interfere with each other. This algorithm is also an integral part of the system to protect the confidentiality of information transmitted: the transition occurs at a pseudorandom algorithm and is determined separately for each connection. When transmitting digital data and audio (64 kbit / s in both directions) using different encoding schemes: the audio is not repeated (usually), and digital data in case of loss of the information package will be retransmitted.

Bluetooth protocol not only supports the connection «point-to-point», but also the connection «point-to-multipoint».

### 2.1.2 Wi-Fi

Wi-Fi -is a trademark of Wi-Fi Alliance for wireless networks based on the IEEE 802.11 standard. The acronyms Wi-Fi include a whole family of standards for the transmission of digital data streams by radio.

Typically diagram of Wi-Fi network comprises at least one access point and at least one client. It is also possible to connect two clients in point-to-point (Ad-hoc), when the access point is not used, and the clients are connected by network adapters "directly".

#### Advantages of Wi-Fi

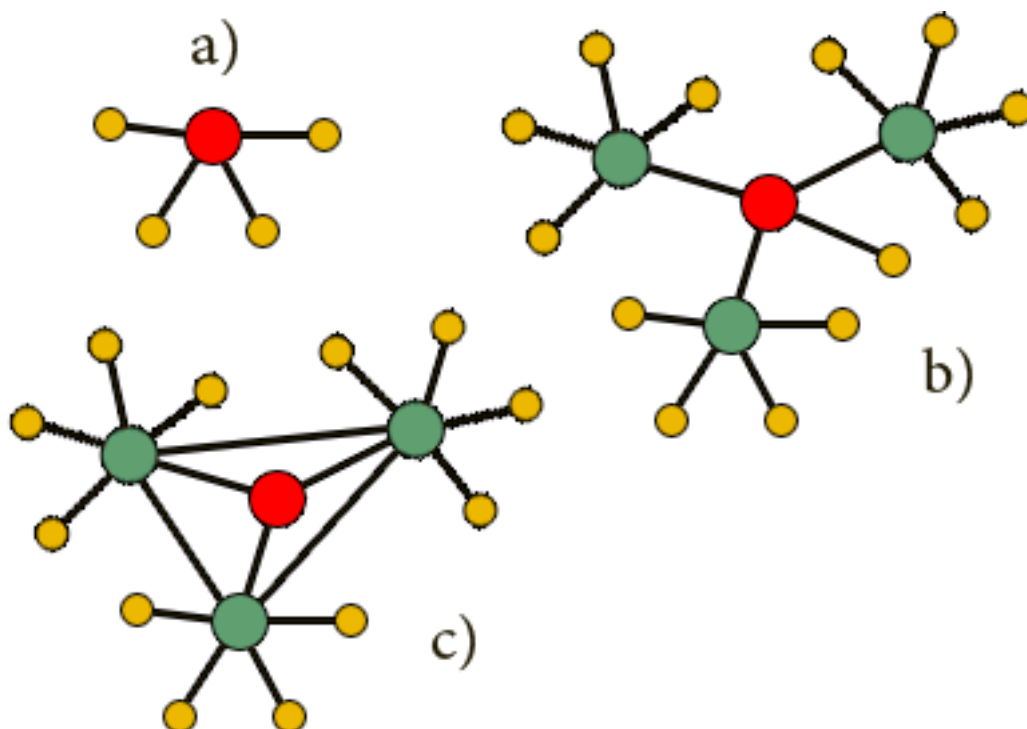
- Allows deploying a network without laying cable, which can reduce the cost of deployment and / or expansion of the network. Areas where cable networks could not be placed, such as outdoors or in buildings of historical value, can be served by wireless networks.
- Allows having access to the mobile devices.
- Wi-Fi devices are widely circulated in the market. Guaranteed compatibility of the equipment due to mandatory certification of equipment with a logo Wi-Fi.
- Mobility.

- Within a Wi-Fi zone to the Internet can go multiple users with computers, laptops, phones and so on.
- Radiation from the Wi-Fi devices when transmitting data to the order is smaller than the cell phone.

### 2.1.3 IEEE 802.15.4

802.15.4 - Specification of IEEE standards regulates the functioning of the physical layer of equipment designed for low-speed wireless personal networks. IEEE 802.15.4 is base for ZigBee technology [20]. IEEE specification aimed at applications that require data transmission at relatively low speeds and the possibility of long-term operation of network devices from independent power sources (batteries).

The main feature of IEEE 802.15.4 technology that with low power consumption it support as simple network topology ("point to point", "tree" and "Star" fig.4), as a self-organizing and self-healing mesh topology relay and message routing. The standard specifies the frequency range, modulation type, the structure of a package, rules of checksum, ways to prevent collisions [21, 22].



**Figure 4. Network topology a) Star, b) Tree, c) Mesh; where red dot – coordinator, green dot – router, yellow dot – end device**

The main areas of application of 802.15.4 technology are wireless sensor networks, automation of property ("Smart Home" and "intelligent building"), medical equipment, industrial systems monitoring and management.

The technology is still at an early stage of development. On the one hand, this technology is supported by leading semiconductor manufacturers, over the years, 802.15.4 specification was substantially revised, and there were public profiles for solutions quite important and popular tasks, such as home automation, energy management, automation of production.

In theory it is possible to build a network of tens of thousands of devices. Network provides address up for 65,535 devices. But protocol specification defines a number of other parameters (associated indirectly with the maximum number of devices on the network) so that it is really possible to say no more than a few hundred devices on a single network. With devices including more than 300 service traffic increases significantly and, accordingly, decreases the bandwidth. The main limiting factors for a large number of nodes - is the cost of resolution of conflicts that arise at random address allocation, capacity reduction due to the growing service activity, long delays in device discovery, and routing restrictions RAM to store large routing tables.

Data transfer rate on the radio channel is 250 kbit / s, but in spite of this, the real data rate can be significantly less. The main problem arises when trying to transmit sound through network - emerging delays are inevitable and unpredictable. The magnitude of these delays up to several seconds, which makes the transfer of "live" sound totally unrealistic, if we are talking about the mesh-network. Actual speed up to 115 kbit / s can be reached in case of transmitting the audio stream in digital form.

The maximum reasonable number of hops the packet in IEEE 802.15.4 network should not exceed 10-30 hops. Considering that outdoor maximum permissible power 10 mW, network can cover only a small area of the district.

Despite the fact that protocol can be applied for mobile end devices, it is possible to speak about mobility only conditionally. Time of occurrence of the network device or switch between routers is a few seconds.

## 2.2 Microcontrollers

Appearance of microcontrollers with ARM core made the transition from 8 to 32-bit embedded systems natural and inevitable event. Constantly improve the functional richness of embedded systems causes the growth of capacity requirements used microcontrollers and their programs [23].

However, so far the most popular in embedded systems were 8 bit microcontrollers, which in terms of price / performance were out of competition.

From the existing 32-bit microcontrollers are characterized by a number of advantages:

- Low price;
- Low energy consumption;
- The relative availability of high-quality software development and debugging.

In recent years in the world of 32-bit microcontrollers occurred events, which radically changed the balance of power. Niche exclusive use of 8-bit microcontrollers was under hard pressure from the 32-bit devices.

Development of 16-bit microcontroller goes much slower than 32-bit. This again suggests that when making a decision about changing of 8-bit microcontroller with more powerful it makes sense to consider, first of all, a 32-bit chip, bypassing the 16-bit.

Cost as a function of the level of technology and integration

Massive shift in the semiconductor industry with 0.18 micron technology abruptly reduce the size of the 32-bit crystal and made their cost comparable to the cost of similar 8-bit chip. However, a simple price reduction was not enough. In addition to the cost of the microcontroller on the final price of the whole system is strongly influenced by the cost of the necessary peripherals. In this regard, 8 bit microcontrollers traditionally "give odds" 32-bit any competitor:

- they worked on a single power source
- integrated power terminals and various communication channels
- have built-in reset
- generate system clock frequency from the internal oscillator.

In addition, the most important requirement is a predictable behavior of the system in case of failures (function brownout detector and a watchdog timer) and the presence of FLASH memory protection against unauthorized reading (lock bits).

Appearance of 32-bit microcontrollers, combining the two most important quality - low cost and superior functional richness - making them direct competitors to traditional 8-bit systems.

### **ARM architecture**

ARM abbreviation comes from the name of the English company Advanced RISC Machines, founded in 1990 as a result of cooperation of Acorn and Apple Computer's. The first kernel, called ARM6, was developed in 1991, but the actual implementation was only following the development - the core of ARM7, created in 1993, the Company is not engaged in direct production of chips. Perhaps, this feature has contributed to the amazing popularity of ARM architecture. The company supplies its developments in electronic form, based on its customers design their own microprocessors and microcontrollers.

ARM customers are more than 60 semiconductor manufacturers, among which well-known companies such as Altera, Analog Devices, Atmel, Cirrus Logic, Fujitsu, MagnaChip (Hynix), Intel, Freescale (Motorola), National Semiconductor, NXP (Philips), OKI, ST Microelectronics and Texas Instruments.

At present, the ARM architecture is a leader and covers 75% of the 32-bit embedded RISC microprocessors market. In fact, we can talk about the ARM architecture as an industry standard. Prevalence kernel allows to developers more flexibility to use their own and third-party software developments as transition to the new ARM processor core, as "migration" between different types of ARM microcontrollers.

To decide about transfer to embedded 32 bit platform is necessary to compare the 8 and 32-bit microcontrollers on the following criteria:

- The level of functional integration of microcontroller;
- The cost of the microcontroller;
- System performance requirements and problems of the real time;
- Requirements for power consumption, power conservation;
- The possibility of porting software to a new platform;



- Availability and quality of software development and debugging (IDEs, compilers, debuggers, programmers).

### **Level of functional integration microcontrollers**

A few years ago, 8 bit microcontrollers were out of the competition in terms of functional saturation. Many members of the families 8051, PIC, AVR give the developer the ability to create a whole system almost single crystal.

At the same time, 32-bit microprocessors were presented by some monsters. To create complete systems based on these microcontrollers require a lot of external components: current generators, power monitors, memory chips and the input device output. It was until others company entered in deal - the market leader of 8 bit micro controllers: Atmel, Philips, ST Microelectronics and others. Today we can say that the 32-bit microcontrollers are not only equal, but often outperform their 8-bit counterparts in terms of functional integration.

An additional advantage of microcontrollers based on the ARM core to AVR, PIC, MSP430, is the absence of a monopoly in the design and manufacture of its varieties. More than 60 companies have already acquired the license on ARM core and can offer a lot of varieties of microcontrollers.

In fact, becoming an industry standard ARM core provides developers a rich selection of devices with the broadest functionality that is continuously expanding due to involvement in the development and production of more and more semiconductor companies.

### **The cost of the microcontroller**

Just a few years ago, the concept of "32-bit microcontroller" and "low cost" were absolutely incompatible. The price of the 32-bit device was much more than the price of 8-bit controllers.

First low-cost 32-bit microcontrollers were presented by company Atmel. In 2004, it presents into the market a family of micro controllers AT91SAM7S, which prices start at a few dollars, that is, directly compete with the prices of 8-bit chips.

In 2005, Philips announced the release of microcontrollers LPC2101/02/03, has focused on the so-called dollar-applications market – in wholesale quantities LPC2101 microcontroller price was a little more than a dollar.

In 2006, the company Luminary released a series of Stellaris LM3S1xx, which has a price in the range of one dollar.

From now, the difference in cost is not a determining factor in choosing between 8 and 32-bit microcontroller.

Prices for modern 32-bit microcontrollers are quite comparable with the prices of their 8-bit competitors.

### **System performance requirements and problems of real-time**

Continuous development of the market place increasing demands of embedded systems. For example, a common task microcontroller control washing machine, until recently, was limited to the functions of a finite state machine on several states, managed by small number of buttons and displays the status of the device on multiple LEDs. Microcontrollers in modern washing machines already have to directly control electric drive, minimize power consumption, control the composition of chemicals to reduce emissions, to display information on the graphic display and maintain a network of communication to communicate with the computer serviced.

To solve this complex of problems performance of 8-bit microcontrollers might have not be enough.

For microcontrollers of real-time systems are fundamentally important it speed of response to external events and deterministic of reaction - i.e. the response time should be within fairly narrow time range, regardless of the condition of the microcontroller at the time of occurrence of an external event.

The largest contribution to the uncertainty of the reaction is making such architectural elements as interim Cache memory. Cache improves the performance of the integrated microcontroller due to background sample instructions and data from a relatively slow memory (e.g., FLASH) in a very fast operating memory, capable of operating on the same frequency with the microcontroller core. However, in case of an

external event when the cache does not contain a necessary part of treatment programs, the response to the event can become very slow, and very difficult to predict the response time.

Therefore, for systems with such requirements the most widely used microcontrollers with cores ARM7TDMI, ARM7TDMIS, which do not have the cache memory and their responses on interrupt did not differ from the usual 8-bit microcontrollers.

### **Vector Interrupt Controller**

ARM7 core has only two external interrupt inputs. For real-time systems it is not enough. Therefore, in the microcontroller AT91SAM7S were introduced additional hardware modules. It causes an interrupt vector for controller, which provides the transition to the address of the interrupt handler of one of the 32 interrupt sources with a single transition team. The main idea of it work is that the control unit contains 32 registers of addresses. Each register filled with address of subprogram for each of the 32 possible sources of interrupt.

When an interrupt occurs, the ARM core causes a transfer of control to the address corresponding to the external interrupt. At this address is an instruction that reads data from the interrupt controller, and the controller substitutes the value of the address register, which will correspond to the interrupt.

Thus, the microcontroller goes directly to the desired interrupt processing program without using an additional instruction.

### **Optimizing access to ports of the output**

A typical task of embedded systems is the manipulation of bits on input/output port. The procedure for changing bits usually involves reading the whole word, the imposition of a bit mask and recording a new meaning of the word back to the port. In RISC processors this kind of operation is performed as one command which performs non-breaking series of "read-modify-write."

ARM architecture does not contain commands to provide the cycle memory access. In the presence of multi-tasking interruption may occur between the moments of initial value of the port and recording its modified value. Result of this interruption can change the current state of the port. To avoid this it is necessary to disable interrupts before each bit procedure.

Such procedure is extremely heavy and slow. So, to improve ports efficiency in microcontrollers with ARM core used unusual for 8-bit microcontroller port structure.

Each port is represented as a virtual two registers, one of which serves only setting bits, the other - only for their resetting. As a result, there is no need for a preliminary reading port, disable interrupts and the entire procedure is performed one bit modification instructions.

Microcontrollers with ARM core designed to meet the requirements of real-time systems. Potential problems of effectiveness programs work related to the features of RISC architecture successfully solved by using the original hardware solutions at the level of peripheral modules.

As for energy, then with decreasing technological norms for manufacturing semiconductors gap in energy consumption between 8/16 bit and 32-bit microprocessors rapidly decreases.

Nowadays modern 32-bit microcontrollers with ARM core show results that rivals to popular 8 bit microcontroller families from 8051, HC08, and even such relatively fresh products like ATmega128. They just lose to modern 8 and 16-bit microcontrollers, the leader among which is the family of MSP430, specially designed for applications requiring minimum energy consumption.

Conclusion: to replace the classic 8-bit microcontrollers, may be considered ARM microcontrollers as the unit with the same parameters of energy consumption. If requirements for power consumption an extremely strict it is better to consider other options. For such cases, it is better to use 8 or 16 bit microcontrollers, which significantly superior any ARM microcontroller in terms of energy consumption.

Thus, the requirements for energy savings are the most practical in relation to the 32-bit microcontrollers.

### **2.3 Sensors**

Range finders, is a modern devices are used to measure distances. They used in various sectors of construction and if manufacture of equipment. They are popular due to the fact they provide high precision and range measurements. When measuring long distances their error amounts to millimeters.

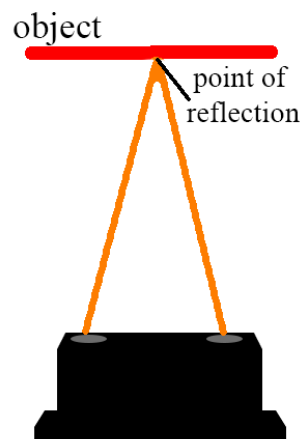
### 2.3.1 Infra-red sensors

#### **Work principles of IR rangefinders**

Rangefinders are used to detect the object in a particular area, as well as to provide data about distance to the object.

To determine the distance or presence of an object in the field of view of the sensor using the method of triangulation and a small linear CCD matrix – special analog integrated circuit, consisting of a light-sensitive photodiodes made of silicon and used CCD technology - charge-coupled devices.

The basic idea of IR rangefinder is next. Pulses of infrared radiation emitted by the transmitter. This radiation propagates and is reflected from objects located in the field of view of the sensor. The reflected light returns to the receiver (fig.5). Emitted and reflected beams form a triangle "emitter - the object of reflection - the receiver."



**Figure 5. Work principle of Infra-Red rangefinder**

The angle of reflection depends on the distance to the object. Received reflected pulses are going to high-quality lens and transmitted to the linear CCD matrix. Based on illuminations on a certain area of CCD matrix the angle of reflection and the distance to the object are calculated.

This method is more protected from the effects of radiation and interference of different reflectivity surfaces made of different materials and painted in different colors.

## Normalization of nonlinearity output

Output characteristic of rangefinders depends nonlinearly on the measured distance (fig.6). It is necessary to find function to convert analog output voltage into distance—centimeters or inches.

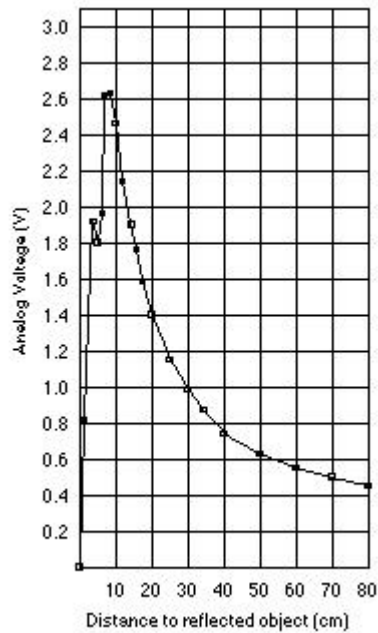


Figure 6. Example of output characteristic

Consider the way of conversion function to find more real values of distance. In solving this problem it is necessary to carry out operations with floating-point. Therefore, to simplify the calculations, apply a linear transformation of the output characteristics.

Based on Sharp Device Specification for GP2Y0A21YK graph characteristics should be transformed by the formula (1):

$$V = \frac{1}{R + k} \quad (1)$$

V — voltage, result of MCU calculations;

R — distance;

k — constant value.

This equation gives a straight line. Constant value in function depends on model of sensor. It is necessary to carry out experiment to find this constant

Next step – it is find a linear characteristic in form of linear equation (2):

$$y = m \cdot x + b \quad (2)$$

In this case  $y$  correspond to distance, change  $y$  и  $x$ :

$$\frac{1}{R + k} = m \cdot V + b \quad (3)$$

Convert it to obtain function for distance on voltage (4):

$$R = \frac{1}{m \cdot V + b} - k \quad (4)$$

This result contain operations with float-point, which can be removed:

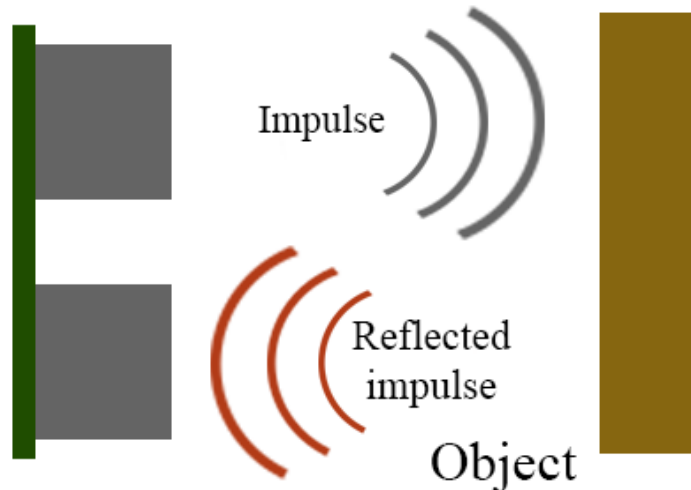
$$R = \frac{m'}{V + b'} - k \quad (5)$$

Where  $m' = \frac{1}{m}$  and  $b' = \frac{b}{m}$ . Now formula can be applied in integer calculations [24].

### 2.3.2 Ultrasonic distance measurer

#### **Work principle**

Work principle of the ultrasonic range finder based on the emission of ultrasound and its reflection from the front are objects (fig.7). Based on the time of sound return it is possible to calculate the distance to the object.



**Figure 7. Principle of work of ultrasonic rangefinder**

Whole work can be divided into 4 steps:

1. Send pulse duration of few second on Trig pin.
2. Inside rangefinder input pulse converted into 8 pulses with 40 KHz frequency and sending through T “eye” of device.
3. When pulses will reach the obstacle, they would be reflected and received by R “eye” of device. This signal called Echo.
4. Controller convert received signal into the distance by formula:

$$S = \frac{F}{58} \quad (6)$$

Where S – distance in centimeters, F – length of impulse [25,26].

## 2.4 Conclusion

Summarizing all the above can draw the following conclusions:

- To send data packets of small size and to increase the battery life of a node should use devices based on IEEE 802.15.4 technology.
- To reduce energy consumption of node, but without a significant reduction in computational power, in conjunction with the Xbee devices, it is recommended to use 32-bit microcontrollers ARM7 TDMI-S. They are specially designed to meet the needs, where one of the determining factors is the uptime of node.
- Data obtained from sensors require further mathematical processing, since their raw values hard to understand the person. For these purposes may be used the above formula. However, the most appropriate option would be to carried out a series of



experiments and, with the help of modern mathematical packages, get depending function to the distance from the values of sensor.

- Since the task of further processing continued flow data requires a lot of computing power, it should not be done on the end nodes. It should be performed in the center of monitoring and control. Since the nodes do not need to "visually" understand the data that their received, the values from sensors should not be converted in distance. This will improve performance of the node.

### 3. System design

In this work, network consist two end nodes and one main control center. End nodes present like an autonomous vehicle with certain sets of components. Control center presented as PC with terminal program and/or Matlab application with GUI.

Each node should be controllable: it is necessary to obtain information from these nodes and send it on main control center for further calculation and perform command gained from control center. For this purpose nodes have a microcontroller, which have to be correctly programmed.

Thus, design of the system can be divided on two parts:

- Hardware design
- Software design

#### 3.1 Hardware design

##### 3.1.1 Design of node

End nodes represented by autonomous vehicle. These vehicles have in their equipment set of components which help them to perform their functions – gather data about distance to objects by sensors, process this data with microcontroller, transmit processed information to another device or control center by radio module, chassis to move node at correspond to incoming command.

##### 3.1.1.1 Components

#### LPC H2148

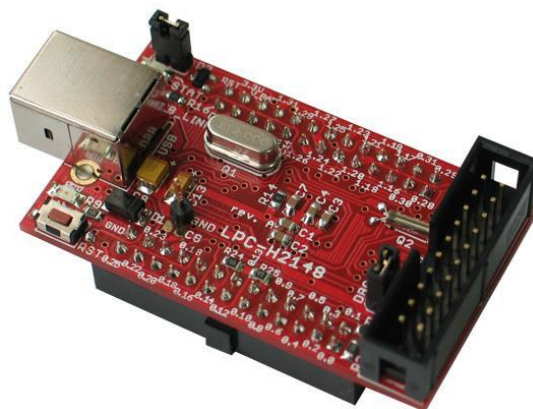


Figure 8. LPC H2148 development board

- ARM7TDMI-S core with supporting of 16-bit set of instruction Thumb;

- Maximum operating frequency – 60 MHz;
- 512 kb FLASH memory with integrated 128-bit bus for faster access;
- in-system programming of FLASH (ISP/IAP);
- 40 kb RAM;
- Integrated resources of JTAG emulator and tracer;
- USB 2.0 Full Speed (12 MHz), 8 Kb RAM for USB data with access through channel of direct memory access (DMA);
- two 10-bit ADC with conversion time of 2.44 ms per channel, 14 analog inputs;
- 10-bit DAC;
- two 32-bit timer/counter (four channel sampling and comparison), the PWM module (6 channels), a watchdog timer (WDT);
- real-time clock with a separate power supply and frequency of 32 kHz;
- universal serial bus: two UART, two I2C, SPI and SSP with buffering and adjusting the length of the data format;
- vector interrupt controller with programmable priorities and vectors addresses;
- built-in frequency multiplier up to 60 MHz with stabilization period of 100 ms;
- integrated oscillator for working with quartz 1 ... 25 MHz;
- Powerdown and Idle power-saving modes;
- individual power management of peripherals to optimize energy consumption;
- Brown-Out Detector (BOD);
- built-in power-on reset;
- single source of supply 3.3 V  $\pm$  10%;
- I/O tolerant to 5 V

Characteristics obtained from [27]

LPC H2148 – is brain of the node (fig.8). Sensors, radio modules, LEDs, driver board – all these components are plugged to LPC via wires and breadboard. Through I/O MCU controls the data transmitting, data receiving, sends signals to activate or deactivate motors, gains information about distance from sensor, processes it information.

## Radio-module XBee - Series 1 (802.15.4)



Figure 9. Xbee radio-module

- 3.3V @ 50mA
- 250kbps Max data rate
- 1mW output (+0dBm)
- 300ft (100m) range
- Wire antenna
- Fully FCC certified
- 128-bit encryption
- Local or over-air configuration
- AT or API command set

Characteristics obtained from [28]

Xbee radio module (fig.9) used to establish wireless connection between several devices and, as a result, create a wireless network with good encrypted transmitting channel, low energy consumption, but low speed. To exchange data it is used UART serial port. Serial port is also used to setup firmware on this device and change parameters of network.

To simplify work with Xbee and UART serial adapter is used (fig.10).



**Figure 10. Xbee FT232RL USB to Serial Adapter**

## **Sensors**

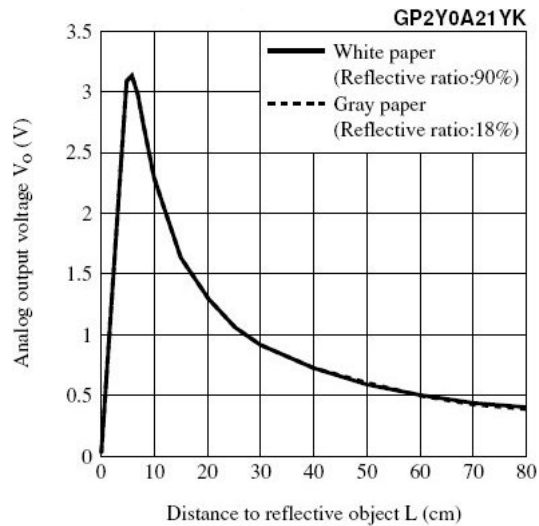
Sharp Infra (IR) Sensor GP 2Y0A21 (fig.11)



**Figure 11. SHARP Infra sensor**

- Energy supply: 4.5 - 5V DC [29]
- Effective angle:  $<40^\circ$
- Measurable range (fig.12) : 10–80 cm

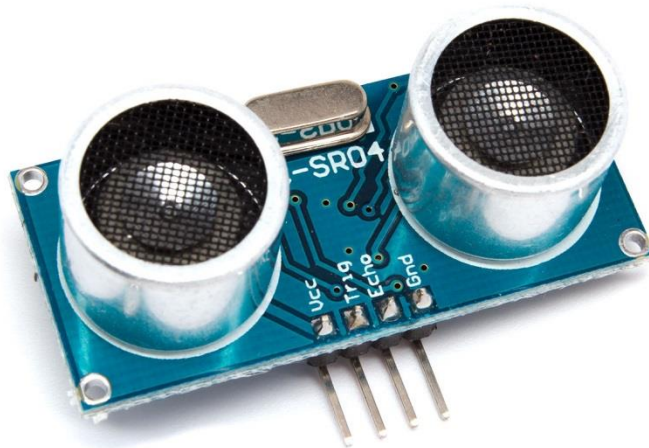
3.1V - 0.4V



**Figure 12. Factory graph characteristic of IR sensor**

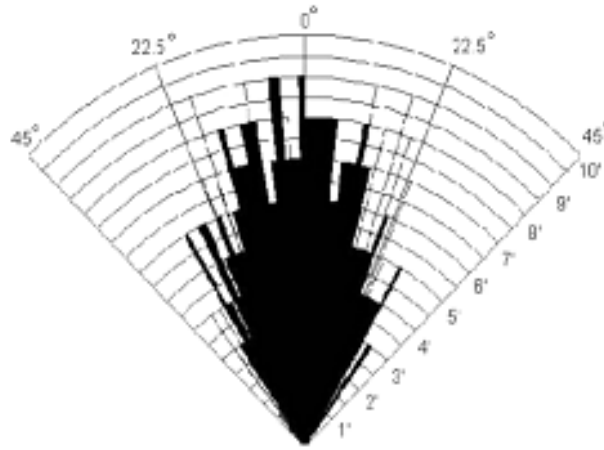
Infra-red rangefinder used to find distance to the object to prevent collision between vehicle and obstacles. Chosen sensor is used under conditions when node works indoor when illumination spreads evenly. Sensor is poorly suited to determine the distance to the light-absorbing objects. In order to determine distances in such an environment better to use ultrasonic rangefinder.

Ultrasonic rangefinder HC-SR04 (fig.13)



**Figure 13. Ultrasonic rangefinder HC-SR04**

- Energy supply: 5V DC
- Quiescent current: <2mA
- Effective angle (fig.14): <15°
- Measurable range: 2–400 cm
- Dimension: 0.3 cm

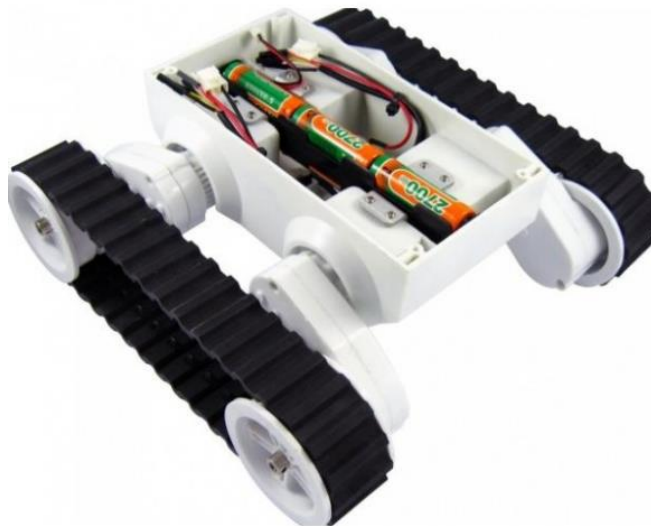


**Figure 14. Depending of measured distance to angle**

Characteristic obtained from [26].

Sonar HC-SR04 is easy to handle. It has 4 outputs: Power, Trig, Echo, and Ground. It is suitable for using outdoor. However, there are difficulties in determining the distance to very small, thin objects.

### **Chasses Rover5**



**Figure 15. Rover5 chassis**

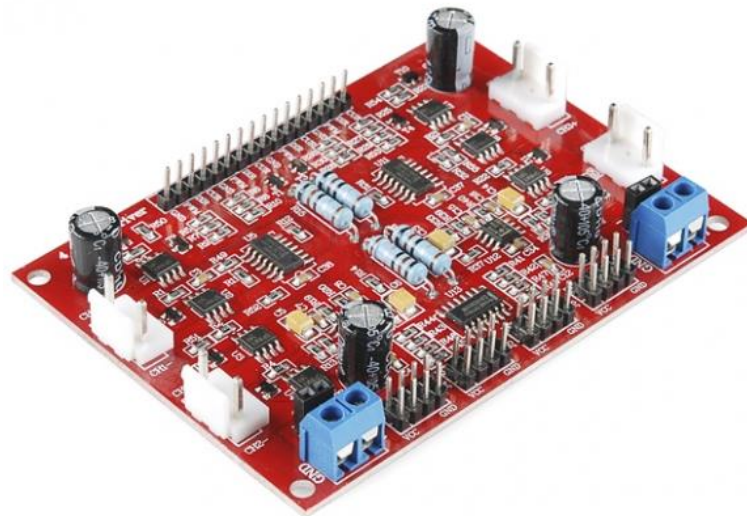
- 4 DC motors
- 4 hall-effect encoders
- Thick rubber tank treads

- 10Kg/cm – torque per motor
- Motors voltage: 7.2B
- Current: 2.5A

Characteristic presented in [30].

Rover5 – robot chassis (fig.15) used as moveable base for different systems. In this work it serves to give nodes possibility to change their position up to sensors results.

### Driver board



**Figure 16. Motor driver board for Rover5**

Control logic:

Each motor controlled by 2 pins.

Direction pin – run forward or reverse.

PWM pin is used to control the motor speed.

Motor output:

Supply for motors 4.5V –12V

Stall current – 4.5A.

Motor driver board (fig.16) serves to connect motors outputs of chasses Rover5 with controller [31].



### 3.1.1.2 *Composition of node*

When all components are chosen connect it each other according to [31, 32]:

- Pin 0.8 and 0.9 of LPC H2148 development board connect to transmit and receive outputs of Serial-to-USB board correspondingly. For this purpose breadboard and wires are used.
- Xbee radio module placed into serial-to-USB board.
- Front IR sensor connected to LPC2148 controller through pin 0.29
- Back IR sensor connected to LPC2148 through pin 0.12
- Pin 0.21 responsible for pulse width modulation. Through breadboard it connected with outputs on motor driver board which marked as “PWM”.
- Motors wires, which responsible for energy supply, connected to outputs on driver board. Wires of encoders connected to inputs on driver board.
- Pin 0.10 connected to direction plugs of motor driver board through breadboard.

#### Source of power

To make the wheels move uses 6 AA batteries, but also rechargeable batteries up to 12 V can be used. Energy source connected to motor driver board input marked as “Vbat”. Further two wires goes from two “Vbat” outputs and connected with breadboard. Now line responsible for Ground connected with all Ground inputs on the side where PWM mark located and line responsible for Current connected with all Current inputs on same side.

Power of LPC2148 can be carried out from 5 volt source. In laboratory conditions, it is powered by the computer through USB connection.

Power of XBee and IR sensor is provided by development board with the controller. For this, the outputs “pin 3.3V” and “Ground” of LPC2148 connected with two different lines on breadboard. Further, wires connect this lines with correspond outputs on sensor and radio-module.

Scheme of all connections presented in appendix under the title “Composition Scheme”.

### 3.1.2 Design of control center

Control center contain two hardware parts – Xbee radio-module placed into Serial-to-USB shield and personal computer/laptop (fig17). They connected via USB cable. For

receiving data on PC it is necessary to use terminal program or special programs which allow reading data from serial port. The mathematical packet of MATLAB may read out data from the serial port, write down them in an array, and carry out their processing.



**Figure 17. Scheme of control center**

Most PCs and Laptops do not have a possibility to work with IEEE 802.15.4 standard. Xbee radio-module plugged to computer allows PCs work with protocol 802.15.4 and, consequently, receive and transmit packages of information.

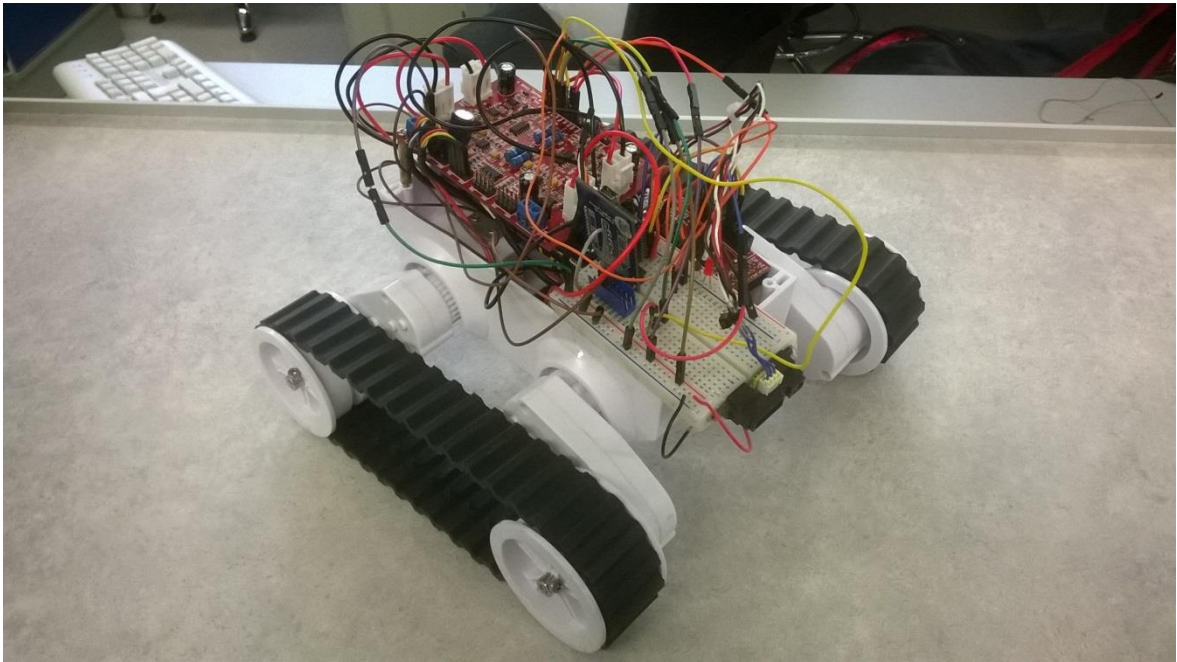
### 3.1.3 Conclusion

All chosen components are ready to use and cooperation each other.

- Xbee radio modules perfect for exchange small packages of data gained from the sensors. Since the nodes would be work in in-door conditions was decided to use infra-red rangefinder. Also, this sensor easier to use because work carried out with only one signal output.
- Computing power of microcontroller will be sufficient to perform all tasks: obtaining data from sensors, processing, transmitting /receiving data wirelessly.
- Rover5 chassis give a node possibility to move from one point to another. Since it has four independent motors, node can rotate left or right. The platform is represented by track base, so it is enough to turn the motors on one side “off”.

As a result, was assembled wheel platform which is shown on fig.18.

Moreover, the system can be improved: transmit data via GSM; add another infrared or ultrasonic sensor to obtain more accurate values of the distance. In all this cases using of other controller board is not required.



**Figure 18. Assembled autonomous vehicle**

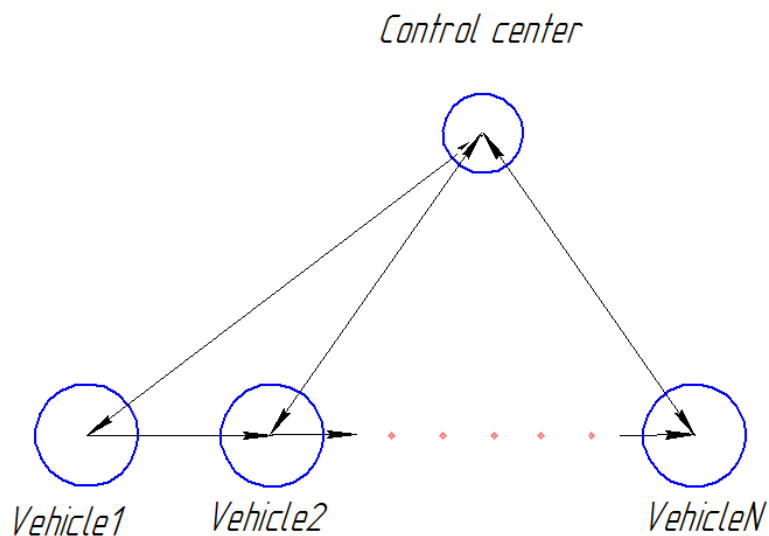
## 3.2 Deployment of network

### 3.2.1 Requires and configuration

To deploy the network it is necessary to configure radio module. It can be performed with a set of commands that are entered in console of terminal software or with help of the X-CTU software.

Since most actions relative to transmitting data like a definition the end of frame performed on LPC H2148, it is necessary to establish only few parameters of modules and network.

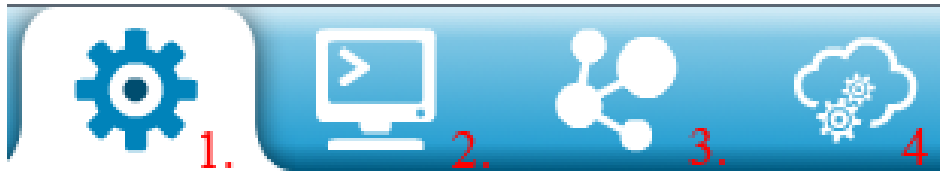
In this system requires that all nodes can communicate each other (fig.19). That is, the need to implement the mesh network topologies. Thus, in the presence of three components – a control center and two wheeled platforms – control center should communicates with the two vehicles, vehicle 1 communicates with the control center and vehicle 2, vehicle 2 communicates with the control center and vehicle 1.



**Figure 19. Network topology**

#### 3.2.1.1 X-CTU

The program is specifically developed by Digi [33] to configure radio modules of their production. The program window contains several tabs (fig.20):

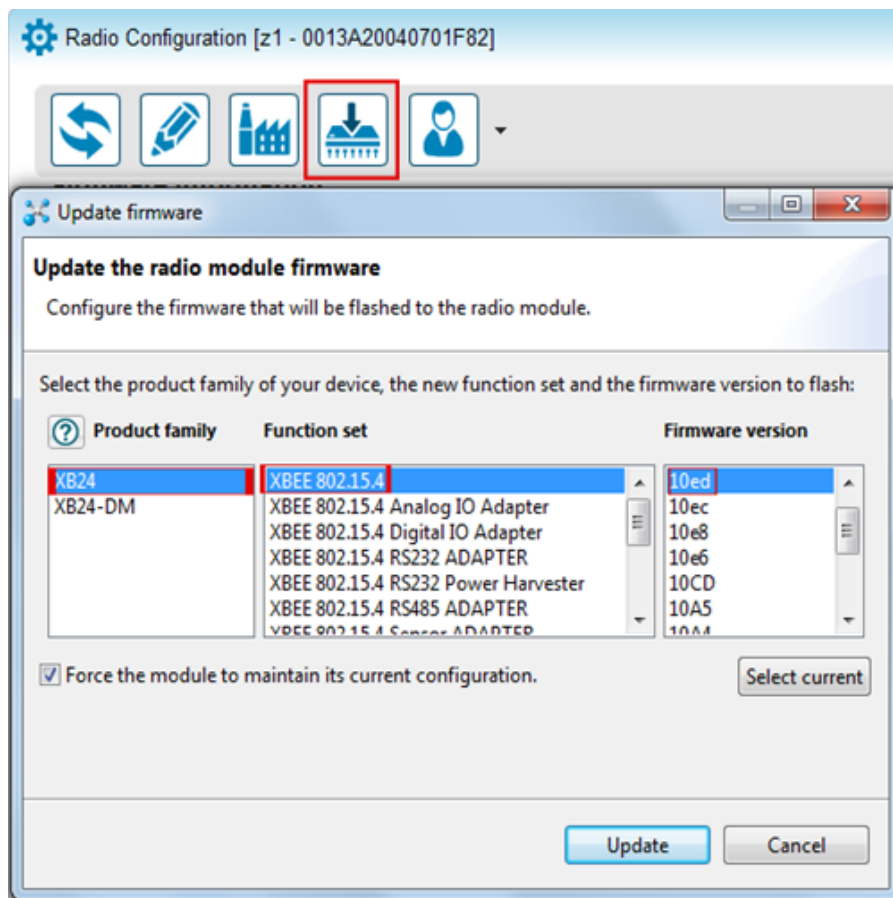


**Figure 20. X-CTU main menu**

1. Configuration of the radio module
2. Console
3. Network test
4. Cloud devices

Then, in the tab "Configuration" of X-CTU software set the following parameters for the radio module of control center:

Should be installed the latest version of firmware for product family XB24 with "Function set" Xbee 802.15.4 (fig.21);



**Figure 21. Update firmware window**

After installing the firmware modify next parameters (fig.22):

PAN ID – identification number of personal area network.

DL – Destination address Low

MY – 16-bit Source Address

#### Networking & Security

Modify networking settings

<b>CH</b> Channel	C
<b>ID</b> PAN ID	6673
<b>DH</b> Destination Address High	0
<b>DL</b> Destination Address Low	1
<b>MY</b> 16-bit Source Address	1

**Figure 22. Setting of the Network ID and Addresses**

DL less than 0xFFFF and DH equal 0. It is mean that for transmitting will be used a 16 bit address.

Since this module is connected to a control center denote it role as a "coordinator" and give it a node identifier "Control center" (fig.23);

<b>CE</b> Coordinator Enable	Coordinator [1]
<b>SC</b> Scan Channels	1FFE Bitfield
<b>SD</b> Scan Duration	4 exponent
<b>A1</b> End Device Association	0000b [0]
<b>A2</b> Coordinator Association	000b [0]
<b>AI</b> Association Indication	0
<b>EE</b> AES Encryption Enable	Disable [0]
<b>KY</b> AES Encryption Key	
<b>NI</b> Node Identifier	Control center

**Figure 23. Setting of the role and name of module**

Leave the other settings as default.

Setting radio-modules of end nodes is performed in a similar manner and with the same parameters. However, the role should remain the "End device", and Node identifier is recorded as "Vehiclen" where "n" – number of node (1, 2, 3...etc.).

### 3.2.1.2 AT command mode

Parameters of network and Xbee device could be established with AT commands, which can be input in some kind of terminal software. Table of all commands presented in Xbee user manual [34].

Thus list of commands to make the same as was done in X-CTU has a next view

+++

AT+CH C

AT+ID 6673

AT+DL 1

AT+MY 1

AT+CE 1 / 0 / 0

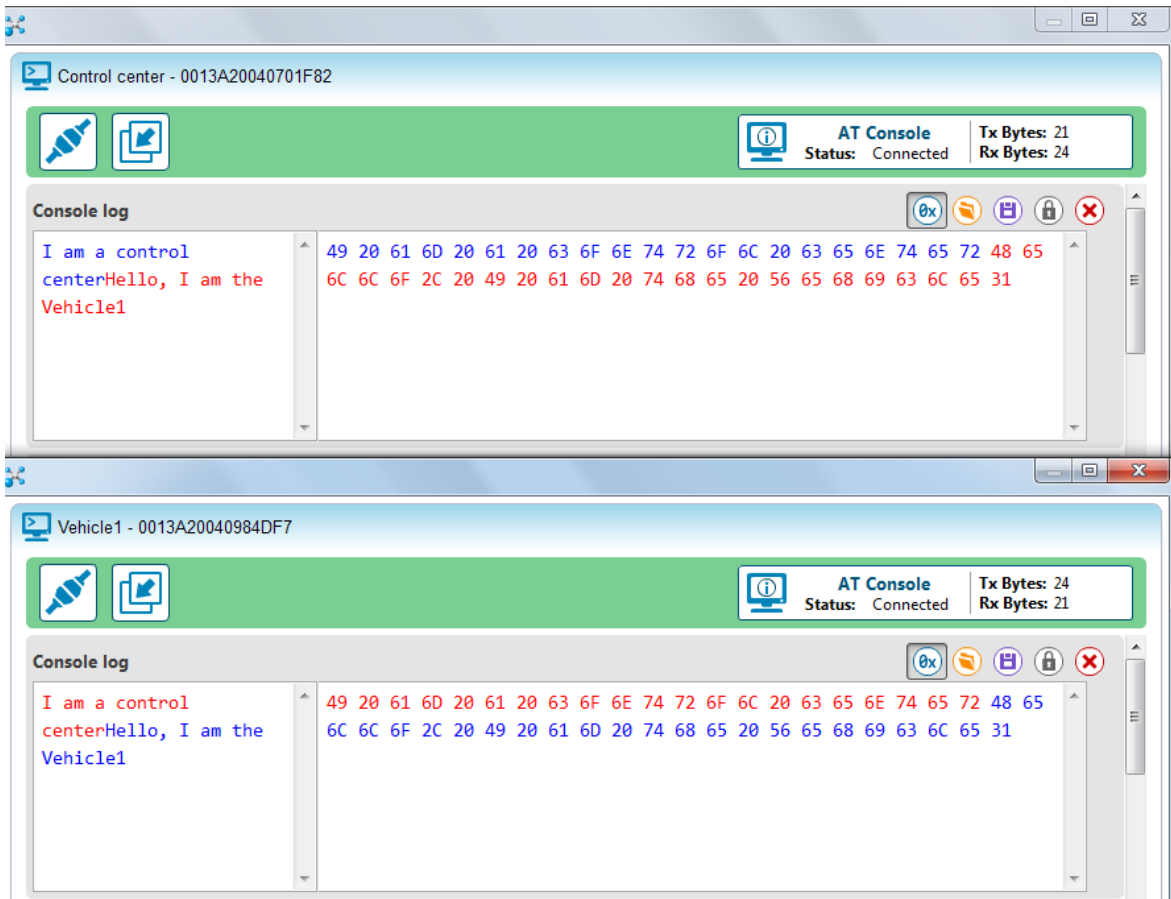
AT+NI Control center / Vehicle1 / Vehicle2

AT+WR

AT+CN

### 3.2.2 Testing the network

To exam network on it possibility to transmit and receive data three Xbee modules were connected to Computer via serial-USB port. In console window of X-CTU utility from Xbee with name “Control center” was sent a frame of data. In case of successful configuration of network and Xbee modules, frame should appear in console windows of other devices. If “vehicle” send a package of information, it also should appear in console window of “Control center” (fig.24).



**Figure 24. Testing of network in X-CTU**

As seen on figure, control center and vehicles communicate each other. It is mean network work correctly.

### 3.2.3 Conclusion

Thus, were established parameters of network and Xbee. Current network has a mesh topology i.e. each node work in broadcast mode and send data on all devices in frame of network with one ID.

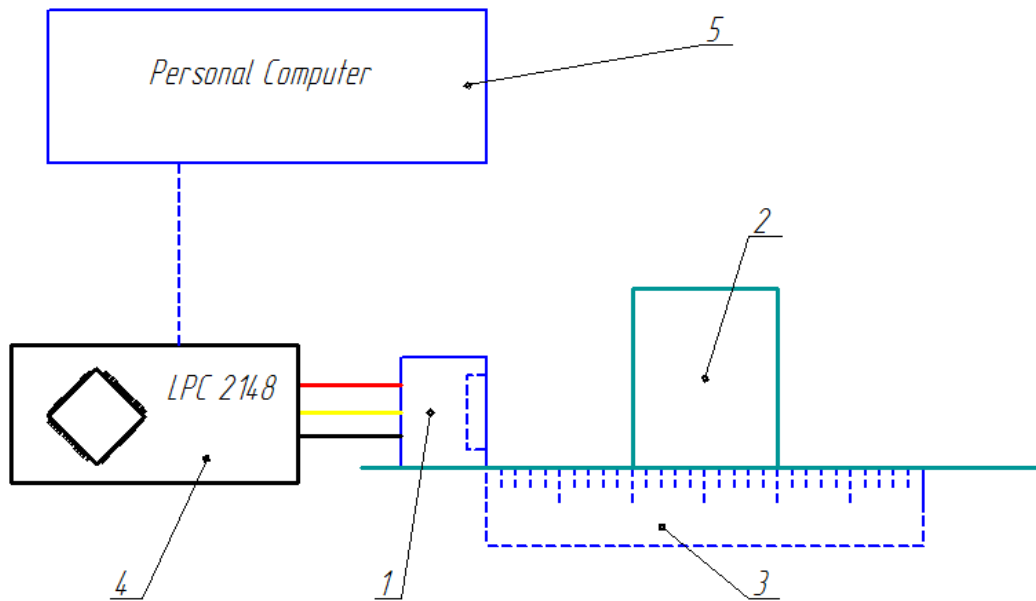
It should be noted, using X-CTU utility make setting of XBee significantly easier, because user do not need to now all commands. Utility has built-in help library and info buttons for each parameter that can be changed. However, X-CTU utility works only with XBee modules made by Digi. Thus, using 802.15.4 radio modules made by third-party companies required the user familiarize with the Xbee user manual which in detail described all AT commands to configure the modules.



### 3.3 Depending function

As was said in Methodology chapter, to convert data collected with sensor into more understandable view it is better to carry out series of experiment. Based on collected results derive a depending function for distance on digital voltage.

Scheme of experiment is shown on fig.25.



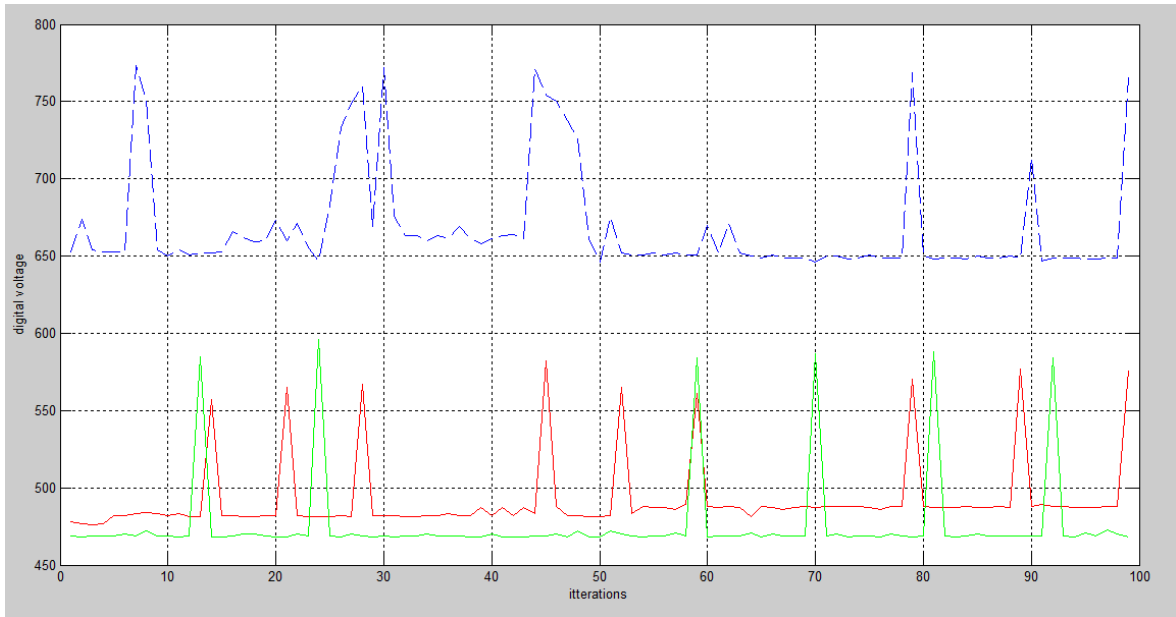
**Figure 25. Scheme of experiment**

Sensor 1 is on the table. In front of the sensor placed obstacle 2. The graph paper or centimeter ruler 3 attached to table. Sensor 1 connected to development board 4, which is connected to the computer 5 via serial port.

Obstacle placed on the certain distance to sensor. PC sends command start to collect data. When one value collected it sends on PC. PC receives these data and put it in array of measured values. When PC carries out one hundred iterations for one distance – data collecting will stop. After that obstacle moves on the next distance and procedure repeated.

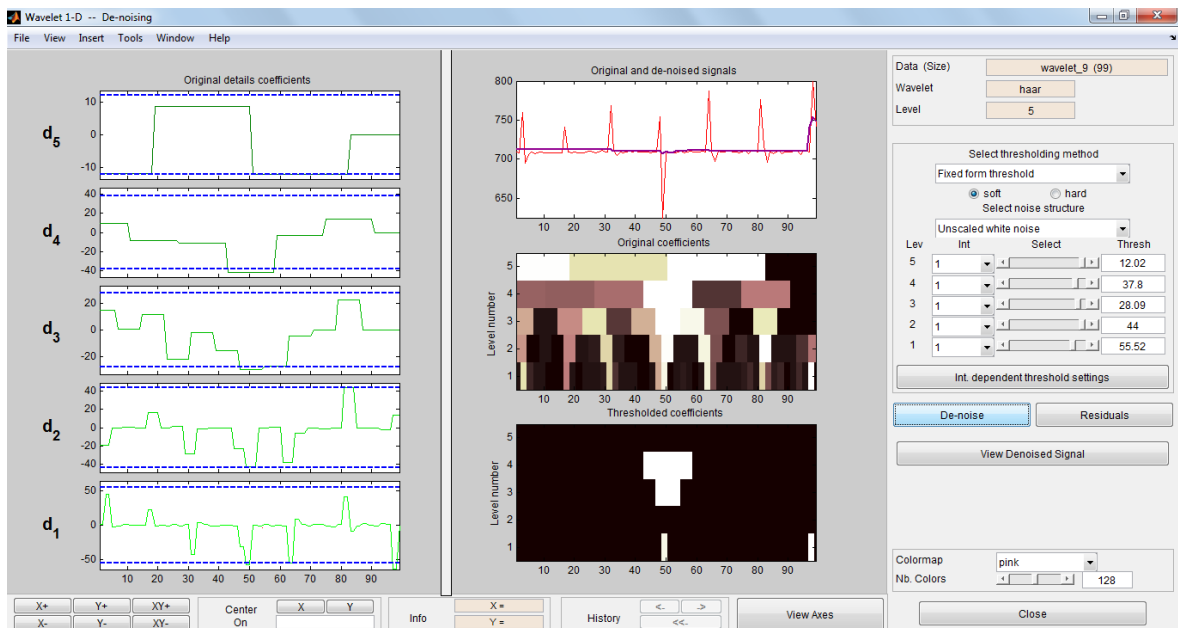
Thus, value of digital voltage for one distance measured one hundred times. Measures for two first distances carried out with step 0.5 cm, last measure – with step 10 cm. All other distances measures with 1 cm step.

All collected signals have a noise (fig.26).



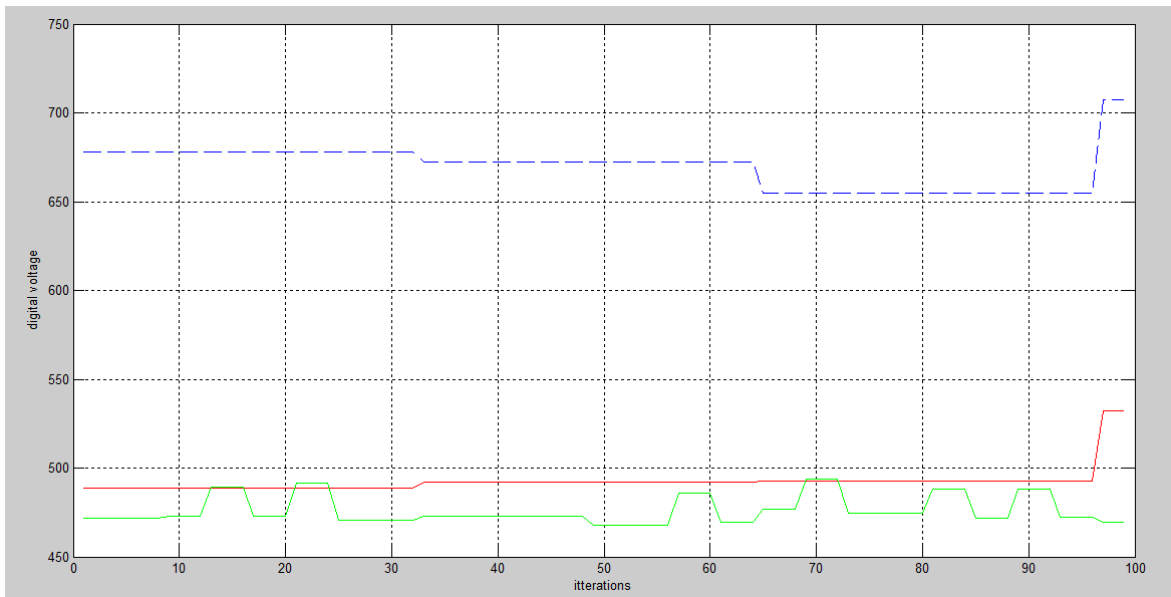
**Figure 26. Set of data for three distance**

To suppress the noise were used Wavelet [35] toolbox of mathematical package Matlab (fig.27).



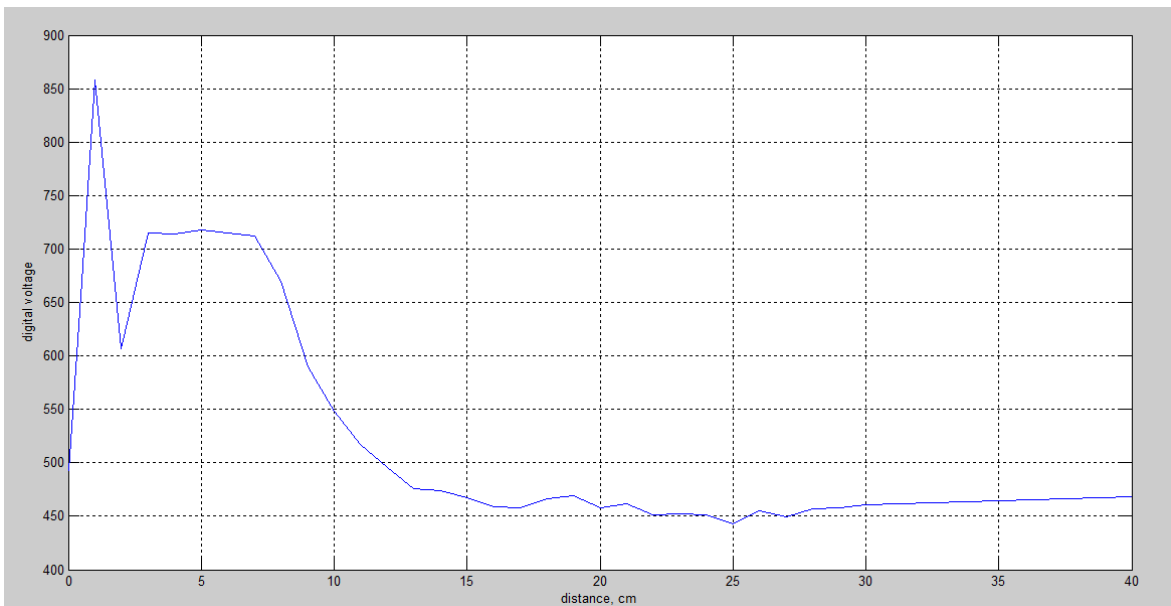
**Figure 27. Wavelet processing**

When collected data was processed it has a view presented on fig.28.



**Figure 28. Data after filtering**

Averaging all values for each distance, it is possible to plot the function of depending of digital voltage on the distance (fig.29).



**Figure 29. Depending of digital voltage on distance**

To find the dependence of the distance of values digital voltage can be used method presented in the chapter «methodology» for infrared sensors. Using that method is complicated by the fact to derivate dependence function coefficients should be selected manually. This method is time-consuming and low precision.

To solve this problem it is proposed to use the method involves finding the coefficients of the polynomial describing the function. To do this, apply the function of Matlab package

```
kofpol= polyfit(meanw(7:33),rng(7:33),3)
```

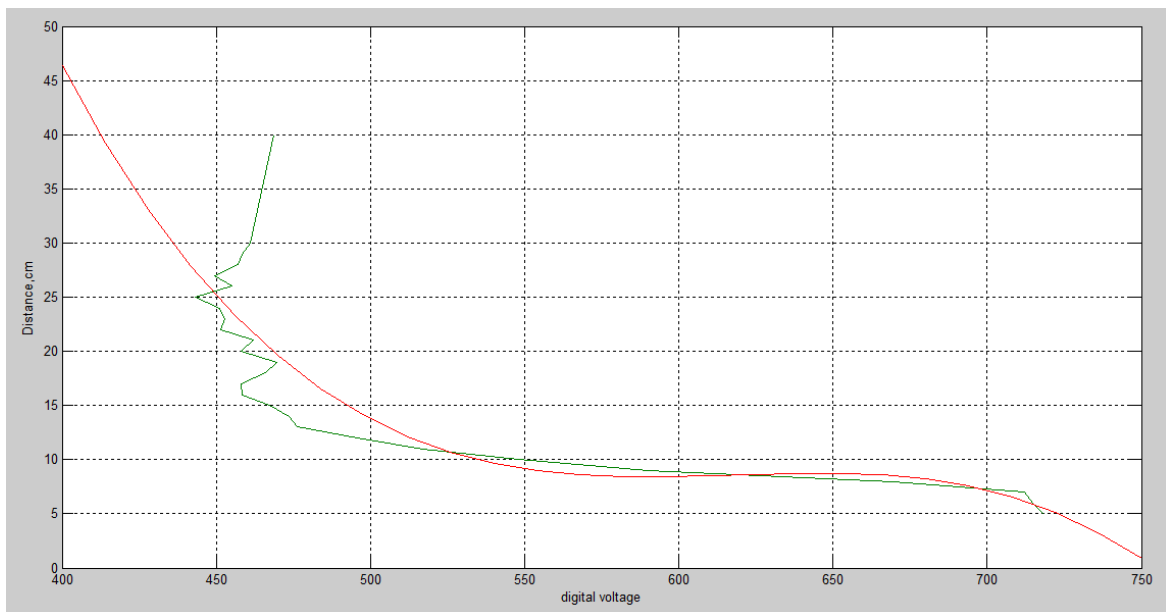
Where meanw – vector of averaged values of digital voltage,

rng – vector of distances

3 – order of polynomial

According to the obtained values of the coefficients plot a function depending on the distance from the digital voltage and compare it with the collected data (fig.30).

```
y= polyval(kofpol,t);
plot(t,y,'--',meanw(7:33),rng(7:33),'+')
```



**Figure 30. Experiment data and polynomial**

As can be seen from the graphs, polynomial relatively accurately describes dependence distance from the digital voltage.

Thus, the function for converting the digital voltage into the distance as follows:

$$R(DV) = -3.86 \cdot 10^{-6} \cdot DV^3 + 7.2 \cdot 10^{-3} \cdot DV^2 - 4.41 \cdot DV + 890 \quad (7)$$

where R – distance in cm, DV – value of digital voltage.

### 3.4 Software design

In this work, it is necessary to design software for nodes and create a program to collect and process data on PC. Thus software design can be divided on next parts:

- Creating executable program for LPC H2148 board with microcontroller to allow nodes perform their functions.
- Creating program to collect and process data on PC with Matlab because need to monitor the state of the system and make corrections to it work

#### 3.4.1 Software for LPC H2148

Designing of software for LPC H2148 carried out in integrated development environment IAR on embedded C. Flashing and debugging of development board performed using JTAG interface and J-Link programmer (fig.31).

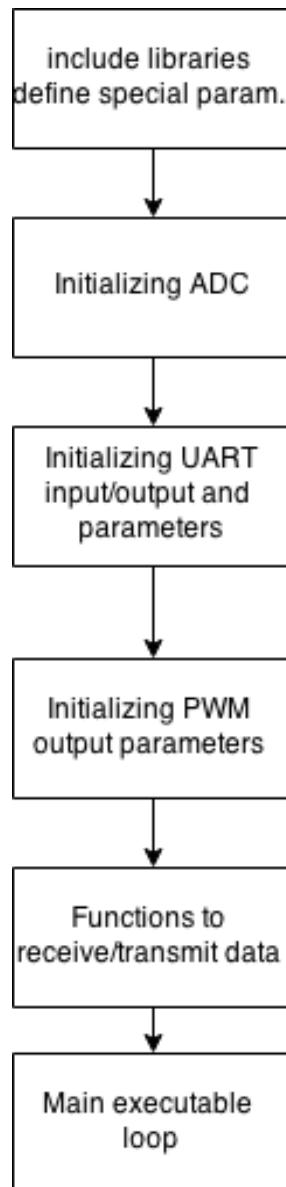


**Figure 31. J-Link EDU Segger**

For writing of software for LPC 2148 it can be used Keil development environment as it was done in [36]

Principle flowchart of program is presented on figure 32. First of all libraries for work with Boolean data type, mathematical operations, operations with strings, work with inputs and outputs and third-party file for working with inputs and outputs of LPC2148 should be linked to this project.

All address of registers, their possible values and executable functions presented in [36].



**Figure 32. Principle flowchart of node program**

Further were defined parameters for Boolean operation, assigned names to events relates to working of motors, established baud rate for UART channel, specified frequency of microcontroller.

After that, initialization function for analog-digital converter was written. This function contain: define pin 0.29 as ADC pin, enabling continuous convert mode, power up the ADC, and starting first conversation.

<pre> AD0CR_bit.SEL = 4; AD0CR_bit.CLKDIV = 12000000 / 4500000; </pre>	<pre> enable AD0.29 only Peripheral Clock (PCLK) 12MHz over ADC sampling rate 4.5MHz (should be less or </pre>
--	--

```

AD0CR_bit.BURST = 1;

AD0CR_bit.CLKS = 0;
AD0CR_bit.PDN = 1;
PINSEL1_bit.P0_29 = 1;
AD0CR_bit.START = 0x0001;

```

equal to 4.5. We generate  
 2.667MHz)  
 put A/D into continuous convert  
 mode  
 11 clocks/10 bit accuracy  
 power up the unit  
  
 start 1st conversation  
 immediately

Now it is necessary to create initialize function for working with UART protocol.

```

PINSEL0 &= 0xFFFF0FFFF;   Reset P0.8,P0.9 Pin Config
PINSEL0 |= 0x00010000;    Select P0.8 = TxD(UART1)
PINSEL0 |= 0x00040000;    Select P0.9 = RxD(UART1)

U1LCR=0x83;               U1LCR: UART0 Line Control Register
                           0x83: enable Divisor Latch access, set 8-bit word
                           length, 1 stop bit, no parity, and disable break
                           transmission

VPBDIV=0x01;              VPBDIV: VPB bus clock divider 0x01: PCLK =
                           processor clock

U1DLL=DIVISOR&0xFF;      U1DLL: UART0 Divisor Latch (LSB)
U1DLM=DIVISOR>>8;        U1DLM: UART0 Divisor Latch (MSB) .
U1LCR=0x03;               U1LCR: UART0 Line Control Register
                           0x03: same as above, but disable Divisor Latch
                           access. And same time U0THR (Transmitting register
                           writing) holding the data

U1FCR=0x05;              U0FCR: UART0 FIFO Control Register 0x05: Clear
                           Tx FIFO and enable Rx and Tx FIFOs

```

Initializing and installing parameters for PWM output to control the motors

```

IOODIR |= (1<<21);        Set P0.21 as output
PINSEL1 &= 0xFFFFF3FF;   Reset P0.21 Function
PINSEL1 |= 0x00000400;    PWM5 - Select pin21 as output for PWM
PWMLER |= 0x00000020;    Enable Shadow Latch For Match-5 (PWM)
PWMMR0 = 1200000>>4;     Set Period = 20mS(PCLK=60MHz) or set PWM
                           Frequency with a cycle of 20mS

PWMPCR &= 0xFFFFFDF;     PWMSEL5 = "0" = Single Edge Control
PWMPCR |= 0x00002000;     PWMENA5 = "1" = Enable PWM5
PWMTCR = 0x00000002;     Reset Counter and prescaler
PWMTCR = 0x00000009;     Enable Counter and PWM + Release Counter from
                           Reset

```

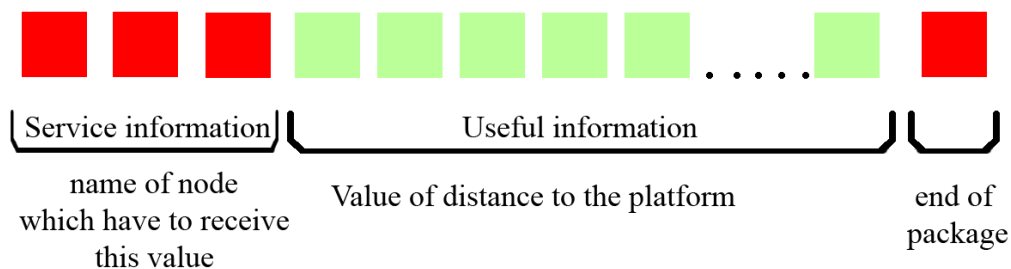
When all necessary elements were initialized and installed it is time to write function for receiving the data. Sampled physical performance of UART presented as RS-232 plug with 9 input/output. In this work, signals carrying hardware presented as 4 pins device, where only 2 output work with transmitting and receiving data. It is mean that other outputs should be emulated by software.

Thus, it is necessary to add service information in data by yourself.

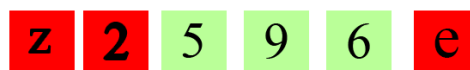
Then function of receiving of string of data is faced with the following tasks:

- To recognize that data aimed on current device
- To write down symbols in string
- To recognize end of package

Each message is encoded so that the structure has two parts: a useful and service. Useful part contains recorded measured value of the digital voltage collected from the sensor connected from behind of platform. The service part – is address of the device-receiver and end of package mark (fig.33). Thus, it is possible to trace the signal values received at a particular device.



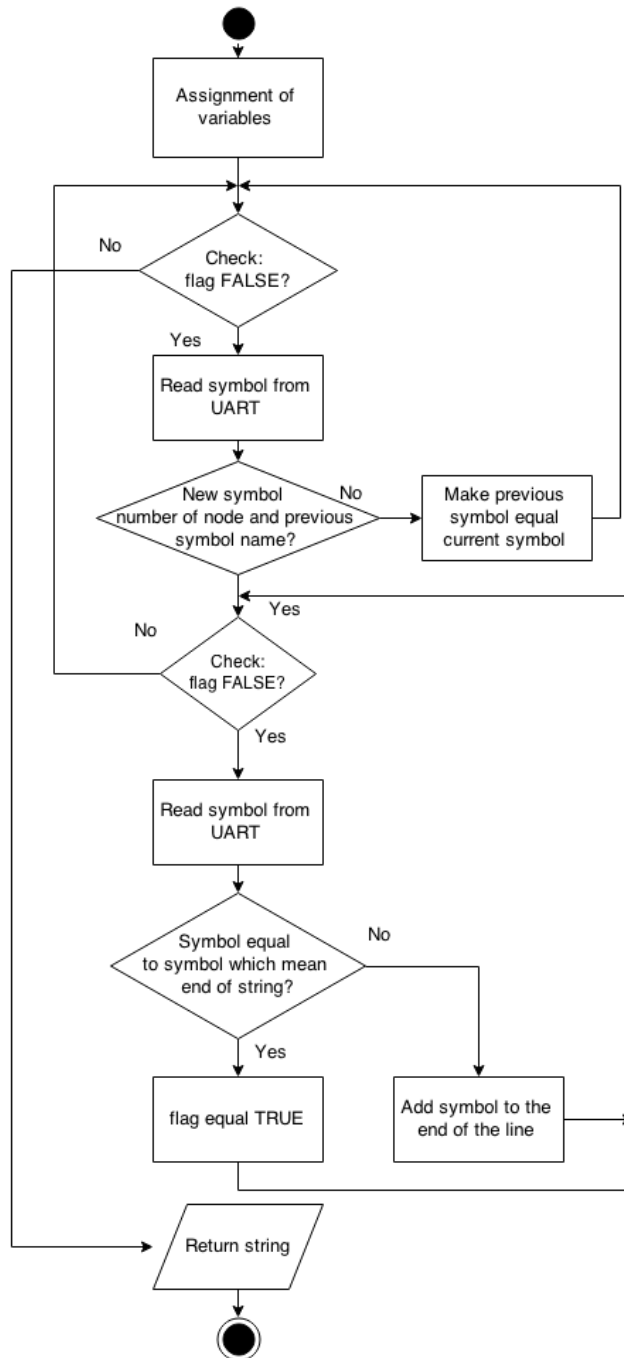
### Example



**Figure 33. Package of data**

Address has a next form: “znRNGe”, where n – number of target node, RNG – value of measured distance, “e” – symbol indicated end of message. Receiver finds these tags in flow of data and save string in variable for further processing. Flowchart of receiving function presented on figure 34.





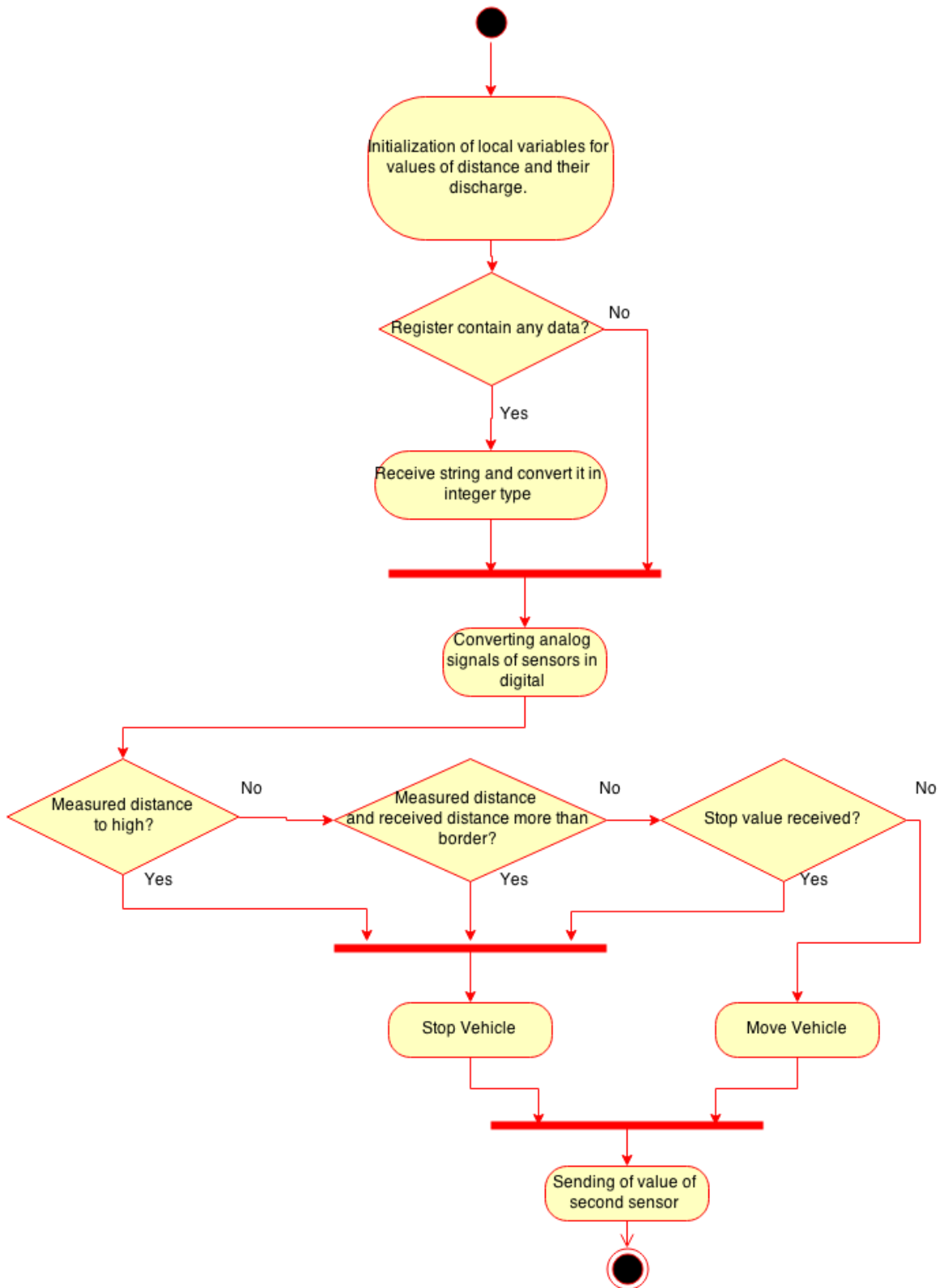
**Figure 34. Flowchart of function for gathering string of data**

Now consider main loop of program. In this cycle program should collect data from sensor, process it and prepare to send on another device, calculate information coming from the other source, and generate command signals for motors.

First of all should be assigned all inputs and outputs which take part in working of node and about which was written above.

Further applied initialization functions for ADC, UART and PWM

Now start work continuous loop (fig.35).



**Figure 35. Main loop of program**

End part of this program – declare of interrupt function. If interrupt occurs in the code, the core immediately stops executing the code it runs, and starts executing an interrupt routine instead. It is important that the environment of the interrupted function is

restored after the interrupt is handled (this includes the values of processor registers and the processor status register). This makes it possible to continue the execution of the original code after the code that handled the interrupt was executed.

In sum all this parts give a program which allow node to communicate with other parts of network, perform actions on incoming commands and to be the source of this command for other devices.

Full C-code of program presented in Appendix under the “C-code for node” name.

### 3.4.2 Monitoring and control

Main idea of monitoring program – track the distance of autonomous platform other vehicles from behind. Thus, it is necessary to realize next tasks in programs:

- specify the port from which the data is to be read;
- identify the variables that should be tracked;
- develop an algorithm for determining which node data assigned;
- display the data in charts in an understandable to operator form;
- start / stop the movement of wheeled platform.

All the above operations should be issued in the form of a graphical user interface (fig.36).

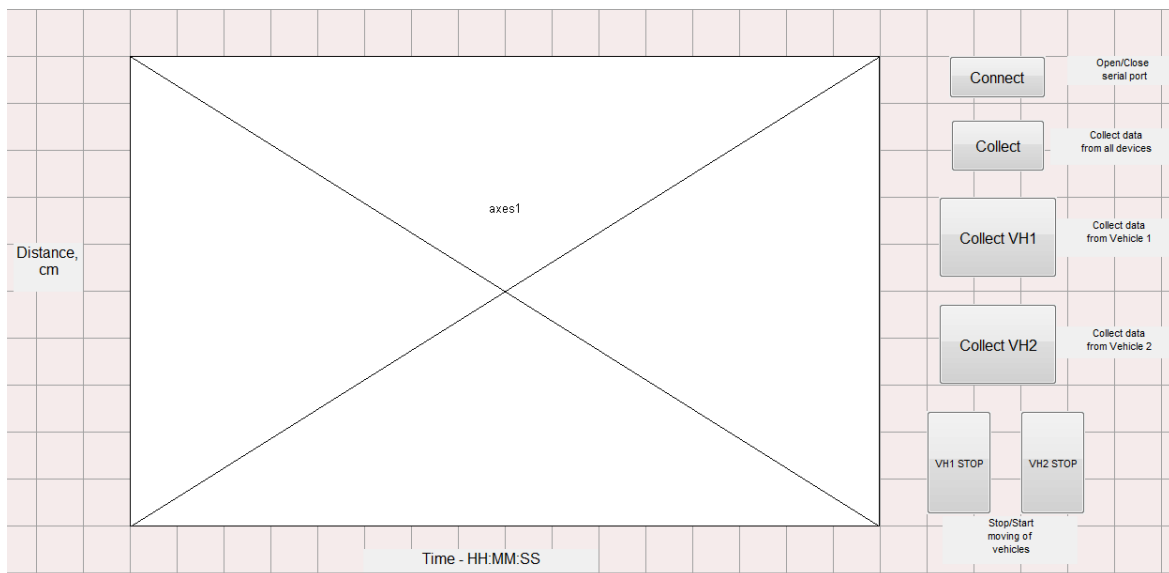


Figure 36. GUI window

Since radio XBee module connected to the computer through the Serial-USB adapter, serial COM port will appear on the computer. X-CTU utility can determine the port number. Knowing the port number, create a serial port object in the Matlab environment and define its properties, such as data rate, connection timeout:

```
s = serial('COM12', 'TimeOut', 1, ...
          'BaudRate', 9600);
fopen(s); %opens the serial port
```

Wheeled platform broadcast digital voltage value. This value - the result of converting an analog voltage obtained from the sensor by analog-to-digital converter of development board. Thus, input value coming from the serial port is a digital voltage.

As it was written earlier, the message is encoded in such way that contains the service characters which should be deleted. The service part will help to define, to what device the message intended, and which array should be used for writing the value.

Program checks which device intended message, symbol by symbol checking name of node and its number.

```
for is=1:length(c)
    if c(is)=='z'
        if c(is+1)=='2'
            n=1;
            for i=(is+2):opa
                if c(i)~='e'
                    abir(n)=c(i);
                    n=n+1;
                end
            end
        end
    end
end
```

Further, the value is processed – digital voltage convert in distance. After processing, a new value is stored in an array related to a particular device.



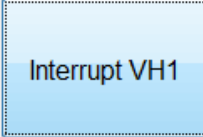
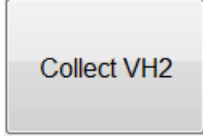

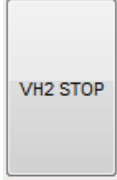
```
a1=str2num(abir);
a1=a1';
ar1=-3.86e-6*a1.^3+7.2e-3*a1.^2-4.41*a1+890;
x1=[x1 ar1];
```

Further, obtained result is displayed in the form of a dynamically changing graph in “axes1” field of GUI.

Monitoring and controlling program also write logs for each node. Logs presented as text files with \*.txt extension. It is contain values of distance and times when these values were received.

```
fprintf(fopen('log1.txt','a+'), '%f\t\t%s\n', ar1, datestr(rem(now,1), 'HH:MM:SS'));
```

In table 2 presented a short user manual for each button of this utility.

№	Button	Function
1		- close the serial port “Connect” state of this button – open the serial port.
2		- obtain data from all devices “Interrupt” state – stop collecting
3		- stop collecting data from Vehicle 1, “Collect VH1” state – start to collect data from Vehicle 1
4		- start to collect data from Vehicle 2 “Interrupt VH2” state – stop collecting data from Vehicle 2
5		- stop the movement of Vehicle 1 “VH1 GO” state – start movement
6		- stop the movement of Vehicle 2 “VH2 GO” state – start movement

**Table 2. GUI buttons**

Full Matlab code for GUI is presented in Appendix under the title “Matlab code for GUP”.

Functions, tools and elements of MATLAB used in development of the interface are considered in [38,39,40,41,42,43].

### 3.4.3 Conclusion

As a result was developed work program for nodes in IAR embedded environment and monitoring and control program for Control center in Matlab.

- Program of nodes allow autonomous platform measure distance to objects and/or obstacles, allow nodes communicate and understand each other, make a decision about movement wheel platform.
- Series 1 Xbee modules communicate with other elements of node with UART interface. Therefore, the microcontroller of development board is programmed to transmit and receive data via UART interface. The algorithm of byte-by-byte reading and comparing of the received symbols allows determine the end of a data packet.
- The program is suitable for all nodes in a network performing the same functions. It is only need to specify the address of the device-receiver and set the address of the current device-transmitter. It should be noted that an increase the order of device numbers, the function of receiving the data string little more complicated.
- GUI for controlling and monitoring has more understandable form for ordinary user than use terminal software. With GUI user does not need to know, how data frame should be formed. In turn, Matlab allow to process data directly after it receiving. It also contains tools and sets of commands for creating a GUI of different complexity. Combination of all this advantages gave a possibility to create friendly graphical user interface to monitor and control actions of vehicles. Operator can choose node by clicking on buttons and check values of one node or all nodes. Incoming data are converted in distance values with centimeter dimension. If it necessary to activate or stop the motors of vehicles operator also can use buttons.

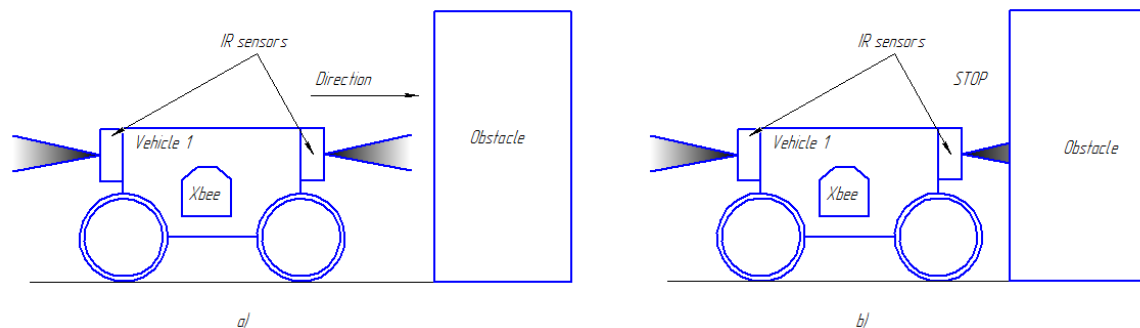
## 4. Results

When all nodes were assembled, network deployed, microcontrollers are flashed with developed program experiments was carried out.

### 4.1 Autonomous behavior

To test autonomous behavior of vehicle when there are no other vehicles in range carry out next experiment:

Put on the way of the vehicle some kind of obstacle. When platform come to close to this object it should stop the motors. Scheme of experiment is present on fig.37.



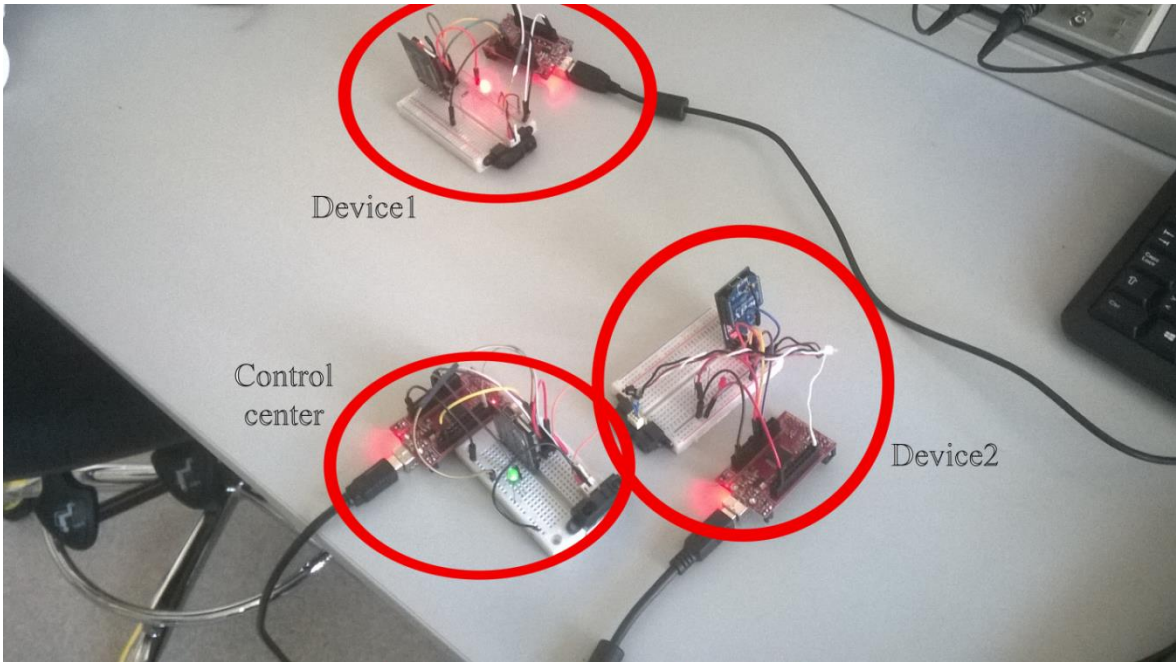
**Figure 37. Scheme of testing of autonomous behavior. a) Vehicle move to obstacle, b) vehicle stopped when reached the obstacle**

### 4.2 Testing of full system

Examine of full system consist three parts. At first, it should be tested communication between nodes, i.e. is one node send data and other receive this data. Secondly, it should be tested, is the PWM was held when one node receive the certain value from the other node or not. Third, test the working of motors and it reaction on changing of received signals.

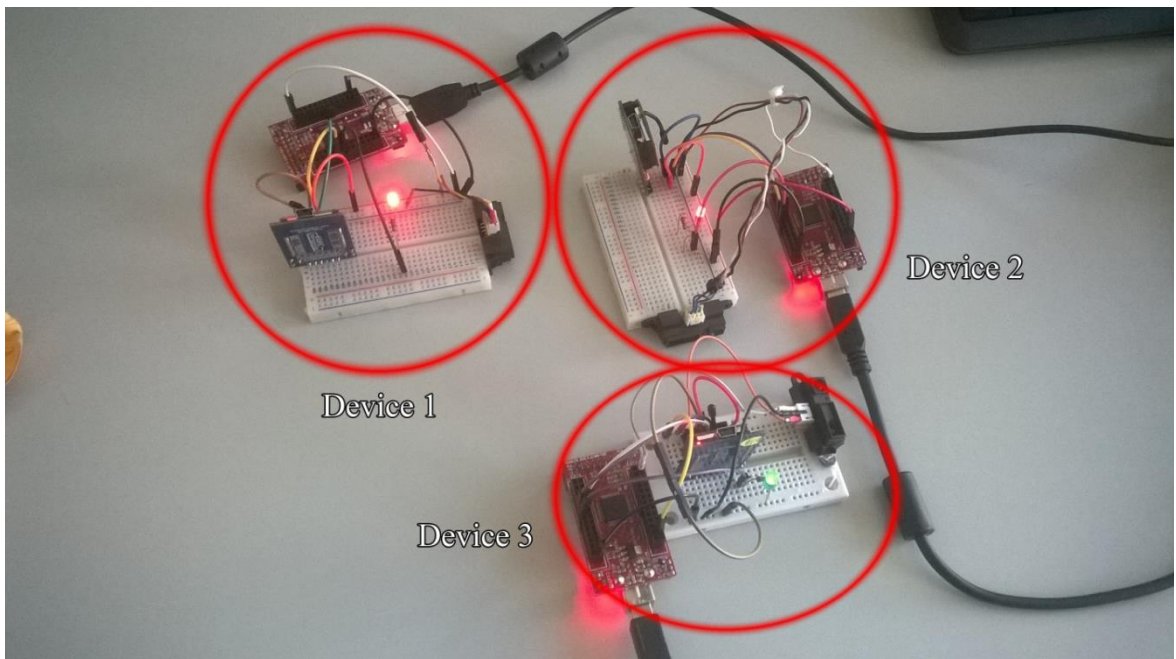
#### 4.2.1 Testing of communication

Xbee radio module connected with development board LPC H2148 as it seemed on figure. Each board also connected to LED and infrared sensor. Main idea of this exam – one node send a distance value collected with IR sensor to concrete node, if other node receive this value and this value require to certain condition – LED goes “on”, else – it “off”.



**Figure 38. Testing network on LPC, pt.1**

On figure 38 evidently device 1 located far from the obstacle, those it send on device 2 the signal, which after processing of the device 2 do not require to condition, so LED off. On the contrary - device 2 is too close to an obstacle and sends back to the device 1 value of the distance that satisfies the LED light.



**Figure 39. Testing of network on LPC, pt.2**



On the fig.39 device 1 to located closer to obstacle so it send value to device 2 which satisfies to condition LED light.

#### 4.2.2 Testing PWM output

To prevent damage to the motor driver board and Rover5 in case of wrong programming of microcontroller and long connection / disconnection of all outputs to the driver board, inspect PWM output using oscilloscope. Controller programed so if received from the other device has a high value – vehicle should stop. It is mean that PWM signal should go to zero. Otherwise vehicle should move forward, i.e. length of PWM signal equal 2.5 ms (fig.40).



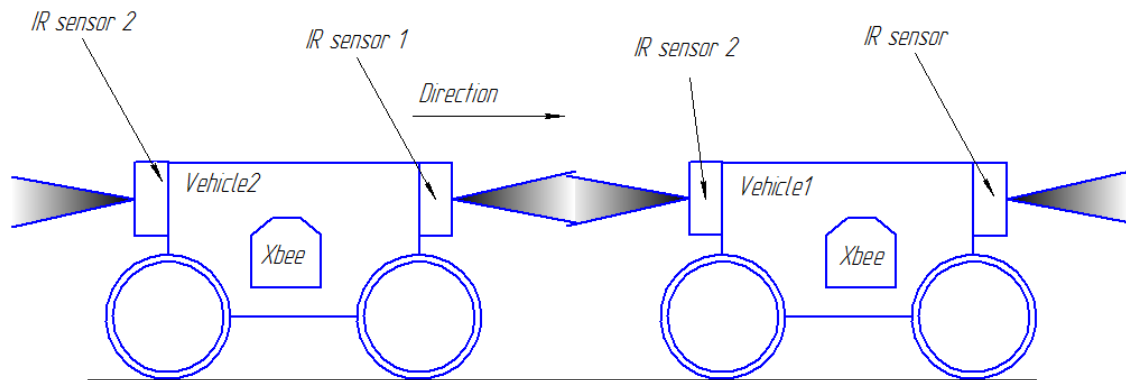
Figure 40. PWM test

### 4.2.3 Reaction of motors

When all outputs of motors connected to motor driver board and pins of this board connected to executive pins of LPC H2148, checked communication between nodes and pulse width modulation, can be tested joint work of all components.

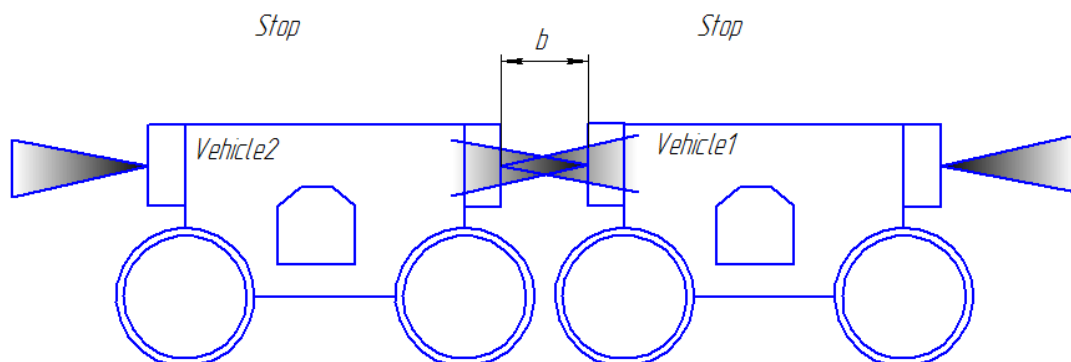
The essence of experiment is follow:

First autonomous system stands on the movement trajectory of second vehicle. Second vehicle moves forward. Vehicles are arranged so that the front sensor of the second platform is looking on backside of the first vehicle and the rear sensor of the first vehicle - on the front side of the second platform (fig.41).



**Figure 41. Test of communication - Initial position**

When vehicle 2 reach the working range of Xbee radio module on vehicle 1, vehicle 2 start to receive values of distance denominated in digital voltage measured by IR sensor 2 of vehicle 1. If received value from vehicle 1 more or equal established border value and value of distance measured by IR sensor 1 of vehicle 2 more or equal established border value vehicle 2 should stop (fig.42).

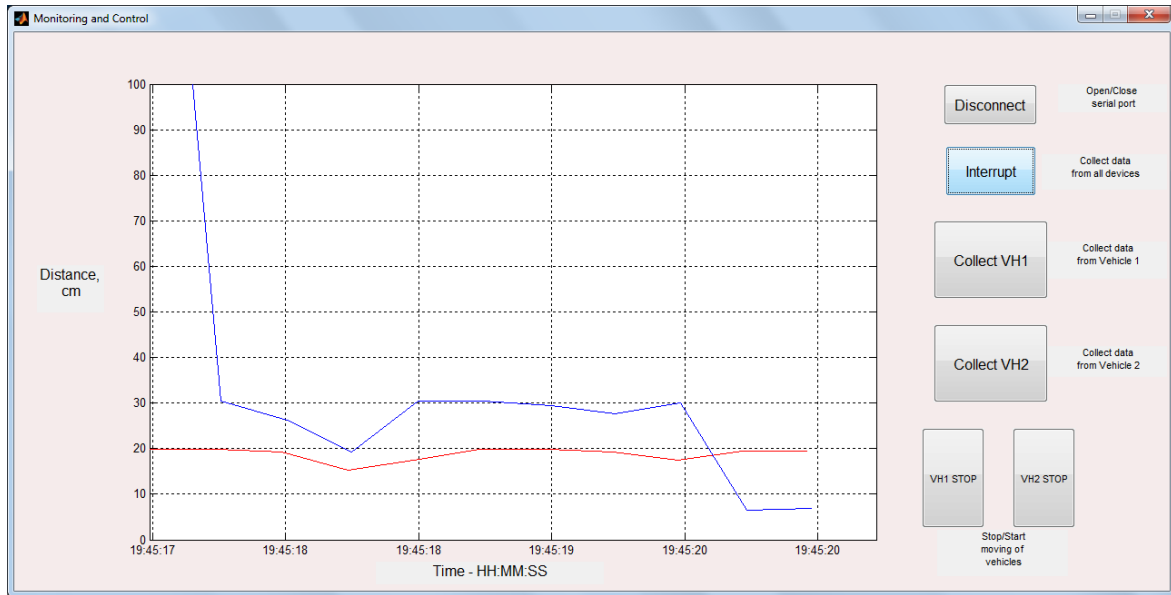


**Figure 42. Test of communication - Stop position**

### 4.3 Testing of GUI

Monitoring and control center serve to track transmitting data and activate or deactivate movement of platform. Thus click on certain buttons should activate commands assigned to this button and vehicles should react on that signals.

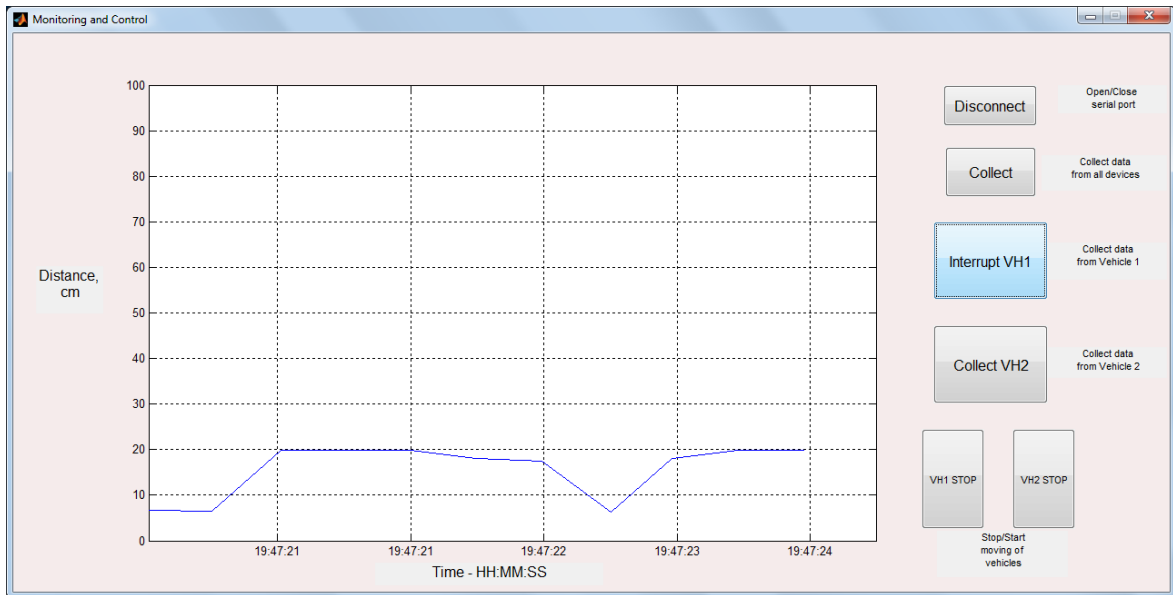
Connection button open a serial port. When port is open “Collect” button start to gather, process, save and plot data from all devices (fig.43).



**Figure 43. GUI button "Collect" test**

“Collect VH1” and “Collect VH2” buttons do the same as “Collect” button, but only for one device.

“VH1 STOP” and “VH2 STOP” buttons serve to deactivate or activate movement of corresponded platform. When platforms in “Stop” position they does not sends data (fig.44).



**Figure 44. GUI button "Collect VH1" test**

#### 4.4 Conclusion

Set of experiments show autonomous platforms execute all their functions which was assigned from the beginning.

If one of these nodes located out of network range it work on program which allow vehicle make a decision about it movement. These decisions based on data collected from front IR sensor.

If wheel platform work in network it cooperate actions with other vehicles in frame of one network. It rely not only the data collected by itself but also on data collected by IR sensors of other devices.

Remote controlling from the control center based of Matlab allow user to track state of system of vehicles and take an actions in case of vehicles going to collide.

## **5. Discussion**

### **5.1 Conclusion**

It has been developed a system of two autonomous platforms and control center, for which was developed the program, based on Matlab, for the controlling of these platforms and tracking of their status.

In assemble used components with low energy consumption. Every single unit is capable of operating for a long time from a single stand-alone power supply. It is worth noting that the sources of power of electronic components and engines of platform should be different. Use of a single source will lead to rapid discharge of node and causing its incapacity.

To allow vehicles communicate with each other and with control center, was deployed a network of 802.15.4 standard. According to the technical documentation for XBee radio modules and set of articles, the data within the network are transmitted at low speed. In practice, however, this fact was not felt. The reason for that – is the small size of the transmitted packets.

Despite the fact that the network has a mesh topology, data packets may be addressed only to a single device. It is provided by structure of package and by algorithms of recognition on end nodes of system. Based on the address, control center keeps track of what data is received by specific node during certain time. Since the data is imported directly into Matlab, they used for processing and are displayed in a user-friendly form and dimension.

To adjust the action, platforms communicate with each other. Despite the fact that platforms are keep the distance between both of them, basing on the data of both machines, it showed the whole viability of the idea of the development of stand-alone devices, the solutions of which would not depend on the presence or absence the control center in network.

Thus, were achieved all tasks set at the beginning: developed a network of wheeled platforms; developed control center with the user interface for controlling of vehicles; developed an algorithm for platforms for working without the control center and / or other platforms.

## 5.2 Future scope

This work is an initial stage of cycle of works on the development of wireless network to control wheeled autonomous systems and their communication with each other. By this, presented development is basis for further network improving, interactions of vehicles and their remote control.

- To improve the reliability of cars it makes sense to add the long-range measuring sensors, duplicate existing sensors for more accurate data, place sensors on the sides of the wheel platform. Web cam plugged to the platform will help operator in control center understand visually, where chosen node located. Adding GPS module to vehicle and exchange coordinates with other nodes prevent collisions between vehicles.
- Encoders installed in the chassis have feedback that allows tracking the position of the output shaft of the engine and, after processing on the microcontroller and / or a control center, to adjust the engine platform.
- As for the possibility of communication, the paper considers the variant when the main load of ensuring signal transmission and receiving falls on the microcontroller, i.e. defining the beginning and the end of the message, node-receiver and so on. The next step is to try to shift some of the obligations on Xbee radio module using the API mode and AT commands to control network. To expand the capacity of Xbee, it makes sense to apply the sleep mode to significantly reduce energy consumption of unit.
- It is also will be wisely to try other wireless technologies and compare the results with used in current work.
- 802.15.4 standard allows to have up to 65 devices within a single network. Stable network operation is feasible with 200 devices in network. Since this work is the basis for further development, work was carried out with only 3 units, 2 of which were endpoints, and 1 - coordinator. The next step is to consider a network consisting of 11 units. In a network with so number of units 1 unit will play the role of coordinator, and 10 others - end devices. Implementation of mesh topology for such number of devices is to be problematic. Therefore it is recommended to start with the implementation of the topology type "Tree" and assign 2 or 3 devices as routers.

- The control center can be presented as personal computer or modern mobile device, such as smart phones, tablet computers based on Android and iOS. With GSM technology and text commands [44], these devices are able to control the platform and track the status of autonomous systems. ARM7 microcontrollers used in the current project have in their kit 2 pairs of UART outputs. One of these pairs is already occupied with Xbee radio module, and the second Tx and Rx pair can be connected to some kind of GSM module.
- Developed in this project GUI based on Matlab allows monitoring and controlling of autonomous platforms. However it contains simple tools for controlling and monitoring: user oriented only on indications of sensors. If vehicle would be equipped with camera, view from the lens should be displayed in GUI window. Data gained from the GPS module help to build trajectory of movement of vehicle in form of graph in GUI window.

## 6. References

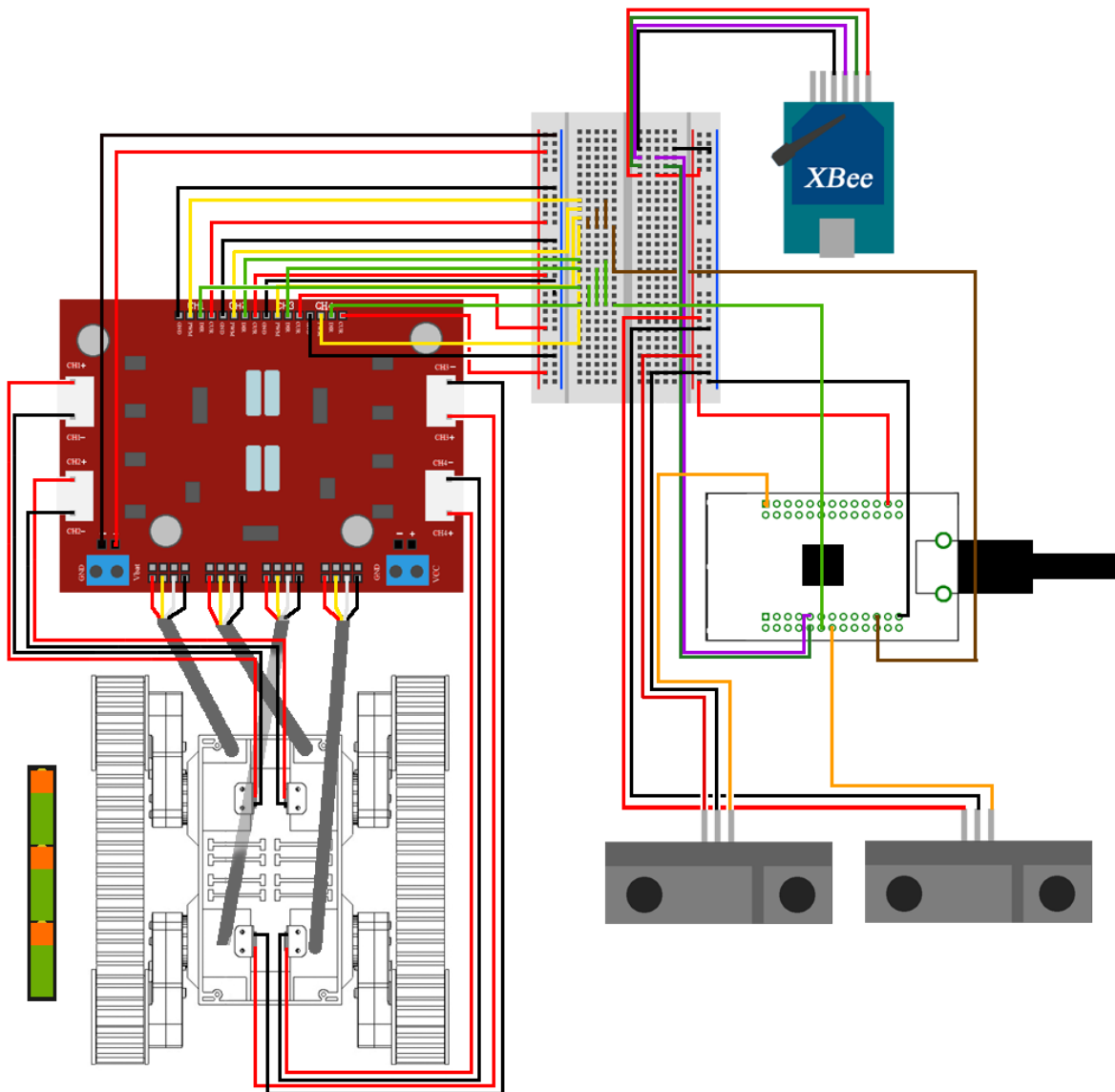
1. <http://www.wikipedia.org>
2. Todd Litman, *Autonomous Vehicle Implementation Predictions: Implications for Transport Planning*, [www.vtpi.org/](http://www.vtpi.org/), 2015
3. Николай Агафонов, *Технологии беспроводной передачи данных ZigBee, BlueTooth, Wi-Fi*, Беспроводные технологии No.1, 2006
4. Michal Varchola, Miloš Drutarovský, *Zigbee based home automation wireless sensor network*, Acta Electrotechnica et Informatica No. 4, Vol. 7, 2007
5. V. Mayalarp, N. Limpaswadpaisarn, T. Poombansao, S. Kittipiyakul, *Wireless Mesh Networking with XBee*, 2nd ECTI-Conference on Application Research and Development (ECTI-CARD 2010), Thailand, May 2010.
6. L.M. Rodríguez Peralta, L.M.P.L. Brito, B.A.T. Gouveia, D.J.G. Sousa, & C.S. Alves, *Automatic monitoring and control of museums' environment based on Wireless Sensor Networks*, EJSE Special Issue: Wireless Sensor Networks and Practical Applications, 2010
7. V. Boonsawat, J. Ekchamanonta, K. Bumrunghet, S. Kittipiyakul, *XBee Wireless Sensor Networks for Temperature Monitoring*, 2nd ECTI-Conference on Application Research and Development (ECTI-CARD 2010), Thailand, 2010.
8. Huang Jinhui, Xue ZiYu, Xu Xu and Tian Siyang, *Zigbee-based Intelligent Home Furnishing*, International Journal of Smart Home Vol. 9, No. 1, 2015
9. C.P.Jeba Samuel, S.Prakash, M. Ananda Kumar, D. Manivannan, *ZigBee Based Home Security Alert System: Integrated Approach*, Journal of Artificial Intelligence 6(1), 2013
10. Vasim K. Ustad, A.S.Mali, Suhas S.Kibile, *Zigbee Based Wireless Air Pollution Monitoring System Using Low Cost and Energy Efficient Sensors*, International Journal of Engineering Trends and Technology (IJETT) – Volume 10 Number 9, 2014
11. Bhagat Singh, Harender, *Intelligent monitoring & controlling of agricultural field parameters using ZigBee*, IJITE Vol.03 Issue-01, 2015
12. Malika, Charu Rana, *A Multi-Alert Patient Health Monitoring using Zigbee*, International Journal of Computer Applications, Volume 83 – No.17, December 2013
13. M. Blaho, J. Ozimy, M. Ernek, M. Foltin, *Zigbee implementation in railroad model*, Mezinárodní konference Technical Computing Prague 2011- Sborník příspěvků, 2011, p.21
14. Mohini Reddy, Vidya Sawant, *Remote Monitoring and Control System for DC Motor using Zigbee Protocol*, International Journal of Application or Innovation in Engineering & Management, Volume 3, Issue 4, 2014
15. Siemens, *Wireless Data Communication via GPRS with S7-1200 and SIMATIC CP 1242-7 GPRS*, 2012
16. Azza Kamal Nabih Hanafi Mahmoud M. Abdallah, *Design and Implementation of ZigBee Based Infrastructure for Smart Environments*, Technical Reprot, Software Engineering Competence Center, 2013



17. AN123 – Breaking the 400-Node ZigBee® Network Barrier With TI’s ZigBee SoC and Z-Stack™ Software, Application Report, 2014
18. Олег Пушкарев, *ZigBee-модули XBee: вопросы практического применения*, Беспроводные технологии №3, 2009
19. Алексей Аникин, *Обзор современных технологий беспроводной передачи данных в частотных диапазонах ISM (Bluetooth, ZigBee, Wi-Fi) и 434/868 МГц*, Беспроводные технологии №4, 2011
20. [www.zigbee.org/](http://www.zigbee.org/)
21. Виктор Алексеев, *Новый ZigBee модуль RC230x фирмы Radiocrafts с поддержкой 6LoWPAN*, Беспроводные технологии №2, 2008
22. Михаил Соколов, Олег Воробьев, *Реализация беспроводных сетей на основе технологии ZigBee стандарта 802.15.4*, Components and Technologies №2, 2005
23. <http://www.phyton.ru/>, Микроконтроллеры ARM7/ARM9 обзор
24. [http://home.roboticlab.eu/ru/examples/sensor/ir\\_distance](http://home.roboticlab.eu/ru/examples/sensor/ir_distance)
25. [www.wiki.amperka.ru](http://www.wiki.amperka.ru)
26. <http://robocraft.ru/blog/electronics/772.html>
27. <http://mail.olimex.com/Products/ARM/NXP/LPC-H2148/>
28. <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#specs>
29. Sharp Infra GP2Y021YK Datasheet
30. Rover 5 chassis brochure
31. Rover5 Driver Manual
32. LPC2148 BOARD HEADER ASSIGNMENTS
33. <http://www.digi.com>
34. Xbee/Xbee-PRO RF Modules, Product Manual
35. А.В.Давыдов, *Вейвлетные преобразования сигналов*, 2004
36. Prashant Kumar, Shashank Singh, *Multisensor system using LPC2148 microcontroller*, National Institute of Technology, 2014
37. NXP B.V., LPC214x User manual, Rev.4, 2012
38. Roddy Posada, Eric Pak, *Arduino, XBee and Matlab GUI Research (HOW TO)*
39. <http://matlab.exponenta.ru>
40. <http://www.cyberforum.ru/matlab/>
41. <http://samoucka.ru/document21551.html>
42. Потемкин В. *Введение в MATLAB СММ*, М.: Диалог-МИФИ, 2000
43. <http://habrahabr.ru/post/195842/>
44. <http://podrugomu.com/node/1379>

# Appendix

## Composition scheme



## C-code for node

```
#include <NXP/iolpc2148.h>
#include <string.H>
#include <stdbool.h>
#include <stdlib.h>
#include <stdio.h>

#define DESIRED_BAUDRATE 9600
#define CRYSTAL_FREQUENCY_IN_HZ 12000000
#define PCLK CRYSTAL_FREQUENCY_IN_HZ // since VPBDIV=0x01
#define DIVISOR (PCLK/(16*DESIRED_BAUDRATE))

#define CR      0x0D
#define TRUE    1
#define FALSE   0

#define DIR (1<<10) //P0.10
#define ADC (1<<12) //P0.12
#define PPM_MAX 120000>>3 //Forward rotating 2.5 ms
#define PPM_MIN 100000>>4 //Backward rotating 0.5 ms

void tiny_delay(void);
void delay_ms(unsigned long int count1);
void UART1_Init(void);
void UART1_Write(unsigned char value);
void UART1_Write_String(char* msg);
int putchar(int a);
char UART1_ReadByte(void);
char* GetStringUSART1(void);
void delay(int cnt);
char* receivedStr;
char buf [500];
char num[10];
int z;
void ADC_init()
{
    AD0CR_bit.SEL = 4; //enable AD0.2 only
    AD0CR_bit.CLKDIV = 12000000 / 4500000; //Peripheral Clock (PCLK) 12MHz
    over ADC sampling rate 4.5MHz (should be less or equal to 4.5. We
    generate 2.667MHz)
    AD0CR_bit.BURST = 1; // put A/D into continuous convert mode
    AD0CR_bit.CLKS = 0; //11 clocks/10 bit accuracy
    AD0CR_bit.PDN = 1; //power up the unit
    PINSEL1_bit.P0_29 = 1;
    AD0CR_bit.START = 0x0001; //start 1st cnvrnsn immediately
}
void ADC1_init()
{
    AD1CR_bit.SEL = 8; //enable AD1.3 only
    AD1CR_bit.CLKDIV = 12000000 / 4500000; //Peripheral Clock (PCLK) 12MHz
    over ADC sampling rate 4.5MHz (should be less or equal to 4.5. We
    generate 2.667MHz)
    AD1CR_bit.BURST = 1; // put A/D into continuous convert mode
    AD1CR_bit.CLKS = 0; //11 clocks/10 bit accuracy
    AD1CR_bit.PDN = 1; //power up the unit
    PINSEL0_bit.P0_12 = 3;
    AD1CR_bit.START = 0x0001; //start 1st cnvrnsn immediately
}
void UART1_Init()
```

```

{
    PINSEL0 &= 0xFFFF0FFF;
        // Reset P0.8,P0.9 Pin Config
    PINSEL0 |= 0x00010000;
        // Select P0.8 = TxD(UART1)
    PINSEL0 |= 0x00040000;
        // Select P0.9 = RxD(UART1)

    U1LCR=0x83;// U1LCR: UART1 Line Control Register
    // 0x83: enable Divisor Latch access, set 8-bit word length
    // 1 stop bit, no parity, disable break transmission
    VPBDIV=0x01; // VPBDIV: VPB bus clock divider 0x01: PCLK = processor
    clock
    U1DLL=DIVISOR&0xFF; //U1DLL: UART0 Divisor Latch (LSB)
    U1DLM=DIVISOR>>8; //U1DLM: UART0 Divisor Latch (MSB).
    U1LCR=0x03 ; // U1LCR: UART1 Line Control Register
    // 0x03: same as above, but disable Divisor Latch access.
    // And same time U0THR (Transmitting register writing)holding the data
    U1FCR=0x05 ; // U1FCR: UART0 FIFO Control Register
    // 0x05: Clear Tx FIFO and enable Rx and Tx FIFOs
}
void PWM_init()
{
    PINSEL1 &= 0xFFFFF3FF; // Reset P0.21 Function
    PINSEL1 |= 0x00000400; // Select pin21 as output for PWM
    PWMLER |= 0x00000020; // Enable Shadow Latch For Match-5
(PWM)
    PWMMR0 = 1200000>>4; // Set Period =
20ms(PCLK=60MHz) or set PWM Frequency with a cycle of 20mS //changed
    PWMPCR &= 0xFFFFFDF; // PWMSEL5 = "0" = Single Edge
// Control
    PWMPCR |= 0x00002000; // PWMENA5 = "1" = Enable PWM5
// Initial Timer Control PWM
    PWMTCR = 0x00000002; // Reset Counter and Prescaler
// PWMTCR = 0x00000009;
// Enable Counter and PWM
//+ Release Counter From Reset
}
int main()
{
    UART1_Init();
    IOODIR |= DIR;
    IOOCLR |= DIR;
    IOODIR &= ~(1<<29)); // Assign input to P0. 29
    IOODIR &= ~(ADC); // Assign input to P0. 12
    IOODIR |= (1<<21); //Set P0.21 as output
    ADC_init();
    ADC1_init();
    unsigned int ADCresult;
    unsigned int ADC1result;
    PWM_init();

    while(1)
    {
        int rec=0;
        if((U1LSR&0x01) != 0)
        {
            receivedStr = GetStringUSART1(); //receiving data
            rec=atoi(receivedStr);
        }
        else;
    }
}

```

```

ADC1result =0;
  ADCresult =0;
  while(AD0DR_bit.DONE == 0);           //wait until conversion done

  AD0DR_bit.CHN = 0;                     // Data in channel 0
  ADCresult = AD0DR_bit.VVDDA;          // save the converted data
  AD0CR_bit.START = 0x0001;
  while(AD1DR_bit.DONE == 0);           //wait until conversion done

  AD1DR_bit.CHN = 0;                     // Data in channel 0
  ADC1result = AD1DR_bit.VVDDA;         // save the converted data
  AD1CR_bit.START = 0x0001;

if (ADCresult>700)
  {
  PWMMR5=0;
  PWMLER = 0x20;
  IO0SET |= DIR; }
else if (ADCresult>520 && rec>520)
  {
  PWMMR5=0;
  PWMLER = 0x20;
  IO0SET |= DIR;
  }
else if (rec==900)
  {
  PWMMR5=0;
  PWMLER = 0x20;
  IO0SET |= DIR;
  while(rec!=200)
  {   rec=0;
    receivedStr = GetStringUSART1(); //receiving data from xbee1
    rec=atoi(receivedStr);}
  }
else
  {
  PWMMR5 = PPM_MAX;
  PWMLER = 0x20;
  IO0CLR |= DIR;
  }
  printf("z3%d%s\n\r",ADC1result,"e");
  delay(900);

  }
}

void UART1_Write(unsigned char value)
{
  /*TEMT (Transmitter Empty)
  TEMT = 0 --> U0THR and U0TSR Contains Some Valid Data
  TEMT = 1 --> U0THR and U0TSR are Empty
  */
  while((U1LSR&0x20) == 0);
  //While U0THR Contains Some Valid Data Stay Here
  U1THR = value;
}

char UART1_ReadByte(void)
{
  /*U0LSR0 is Set When U0RBR Contains Some Valid Data
  otherwise U0LSR0 is Clear*/
  while((U1LSR&0x01) == 0);
}

```

```

        //Stay Here until U0LSR0 set to 1
        return U1RBR;
    }
void UART1_Write_String(char* msg)
{
    #ifndef LENGTH_UART
    #define LENGTH_UART

        int i_uart;

    #endif

        for(i_uart=0;i_uart<strlen(msg);i_uart++)
        {
            UART1_Write(msg[i_uart]);
        }
}

char* GetStringUSART1(void)
{
    static char str[200];
    int i = 0;
    bool done = FALSE;
    char tmp;
    char tmp1=0x00;
    char ends;
    while(!done)
        {
            tmp = UART1_ReadByte();
            if (tmp1 == 'z' && tmp=='2')
                {
                    while(!done)
                    {
                        ends = UART1_ReadByte();
                        if (ends == 'e')
                            {
                                done = TRUE;
                            }
                        else
                            {
                                str[i] = ends;
                                i++;
                            }
                    }
                }
            else
                tmp1=tmp;
        }

    return str;
}

void delay(int cnt)
{
    int i,j ;
    for(i=0;i<cnt;i++)
        for(j=0;j<cnt;j++) ;
}

__irq __arm void irq_handler (void)

```

```

{
void (*interrupt_function)();
unsigned int vector;

vector = VICVectAddr; // Get interrupt vector.
interrupt_function = (void(*)())vector;
(*interrupt_function)(); // Call vectored interrupt function thru
pointer
VICVectAddr = 0; // Clear interrupt in VIC
}

/*****
*
* Function Name: fiq_handler
* Parameters: void
* Return: void
*
* Description: FIQ subroutine
* Note: This is ARM mode code - full 32 bit code
*****/

/
__fiq__arm void fiq_handler (void)
{
while(1);
}

int putchar(int a)
{
while (!(U1LSR&0x20));
return(U1THR=a);
}

```

## Matlab code for GUI

```
function varargout = GUI_new(varargin)
% GUI_NEW MATLAB code for GUI_new.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_new_OpeningFcn, ...
                  'gui_OutputFcn',  @GUI_new_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_new is made visible.
function GUI_new_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for GUI_new
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_new wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_new_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cnt_button.
function cnt_button_Callback(hObject, eventdata, handles)
% hObject    handle to cnt_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if strcmp(get(hObject,'String'),'Connect') % currently disconnected
s = serial('COM11', 'TimeOut', 1, ...
          'BaudRate', 9600);
    try
        fopen(s);
        handles.s = s;
    end
```



```

        set(hObject, 'String','Disconnect')
    catch e
        errordlg(e.message);
    end
else
    set(hObject, 'String','Connect')
    fclose(handles.s);
    fclose(handles.s1);
end
guidata(hObject, handles);

% --- Executes on button press in Get_Data.
function Get_Data_Callback(hObject, eventdata, handles)
% hObject      handle to Get_Data (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
if strcmp(get(hObject,'String'),'Collect')
try
    i1=0;
    i2=0;
    x1=[];
    x2=[];
    d1=[];
    d2=[];
    set(handles.node2Value,'String','Collect VH2');
    set(handles.node1Value,'String','Collect VH1');
    set(hObject, 'String','Interrupt');
    while(strcmp(get(hObject,'String'),'Interrupt')&&...
        strcmp(get(handles.node2Value,'String'),'Collect VH2')&&...
        strcmp(get(handles.node1Value,'String'),'Collect VH1'))
        c =fscanf(handles.s); %reads the data from the serial port and stores
it to the matrix a
        opa=length(c);
        abir='';
        abir2='';
        for is=1:opa
            if c(is)=='z'
                if c(is+1)=='2'
                    n=1;
                    for i=(is+2):opa
                        if c(i)~='e'
                            abir(n)=c(i);
                            n=n+1;
                        end
                    end
                elseif c(is+1)=='3'
                    n=1;
                    for i=(is+2):opa
                        if c(i)~='e'
                            abir2(n)=c(i);
                            n=n+1;
                        %else break;
                        end
                    end
                end
            end
        end
    end
end
end
end

```

```

end

if isempty(abir)==0
    d1=[d1 rem(now,1)];
    a1=str2num(abir);
    a1=a1';
    ar1=-3.86e-6*a1.^3+7.2e-3*a1.^2-4.41*a1+890;
    x1=[x1 ar1];
    set(handles.axes1, 'XTick',rem(now,1)-40.0926e-06:...
8.0926e-06: rem(now,1)+4.0926e-06);
    set(gca, 'XTickLabel',datestr(rem(now,1)-40.0926e-06:...
8.0926e-06: rem(now,1)+4.0926e-06, 'HH:MM:SS'));
    plot(d1,x1,'b');
    hold on; %(x);

fprintf(fopen('log1.txt','a+'),'i\t\t%f\t\t%s\n',a1,ar1,datestr(rem(now,
1), 'HH:MM:SS'));
elseif isempty(abir2)==0
    d2=[d2 rem(now,1)];
    a2=str2num(abir2);
    a2=a2';
    ar2=-3.86e-6*a2.^3+7.2e-3*a2.^2-4.41*a2+890;
    if ar2>80
        ar2=x2(length(x2));
    end
    x2 =[x2 ar2]; % Merging the value to an array, this is not
very computationally effective, as the array size is dynamic.
%Consider pre allocation the size of the array to avoid
this. But beware, You might loose some important
%data at the end!
    set(handles.axes1, 'XTick',rem(now,1)-40.0926e-06:...
8.0926e-06: rem(now,1)+4.0926e-06);
    set(gca, 'XTickLabel',datestr(rem(now,1)-40.0926e-06:...
8.0926e-06: rem(now,1)+4.0926e-06, 'HH:MM:SS'));
    plot(d2,x2,'r');
    hold on; %(x);

fprintf(fopen('log2.txt','a+'),'i\t\t%f\t\t%s\n',a2,ar2,datestr(rem(now,
1), 'HH:MM:SS'));
end
grid on;
xlim([rem(now,1)-40.0926e-06 rem(now,1)+4.0926e-06]);
abir2='';abir='';c='';
drawnow;
end
catch e
    disp(e)
end
else
    set(hObject, 'String','Collect');
    set(handles.node2Value, 'String','Collect VH2');
    set(handles.node1Value, 'String','Collect VH1');
end
guidata(hObject, handles);

% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%if isfield(handles, 's')

```

```

if strcmp(get(handles.cnt_button,'String'),'Disconnect')
    fclose(handles.s);
end
% Hint: delete(hObject) closes the figure
delete(hObject);

% --- Executes on button press in node1Value.
function node1Value_Callback(hObject, eventdata, handles)
% hObject    handle to node1Value (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if strcmp(get(hObject,'String'),'Collect VH1')
try
    x1=[];
    d1=[];
    set(hObject, 'String','Interrupt VH1');
    set(handles.Get_Data,'String','Collect');
    set(handles.node2Value,'String','Collect VH2');
    % while(t~=100) %Runs for 60 cycles - if you cant see the symbol, it
is "less than" sign. so while (t less than 200)
    while strcmp(get(hObject,'String'),'Interrupt VH1')&&...
        strcmp(get(handles.Get_Data,'String'),'Collect')&&...
            strcmp(get(handles.node2Value,'String'),'Collect
VH2')
        c =fscanf(handles.s); %reads the data from the serial port and stores
it to the matrix a
        opa=length(c);
        abir='';
        for is=1:opa
            if c(is)=='z'
                if c(is+1)=='2'
                    n=1;
                    for i=(is+2):opa
                        if c(i)~='e'
                            abir(n)=c(i);
                            n=n+1;
                        end
                    end
                end
            end
        end
        if isempty(abir)==0
            d1=[d1 rem(now,1)];
            a1=str2num(abir);
            a1=a1';
            ar1=-3.86e-6*a1.^3+7.2e-3*a1.^2-4.41*a1+910;
            x1=[x1 ar1];
            set(handles.axes1, 'XTick',rem(now,1)-40.0926e-06:...
8.0926e-06: rem(now,1)+4.0926e-06);
            set(gca, 'XTickLabel',datestr(rem(now,1)-40.0926e-06:...
8.0926e-06: rem(now,1)+4.0926e-06, 'HH:MM:SS'));
            plot(d1,x1,'b');
            hold on; %(x);

fprintf(fopen('log1.txt','a+'),'%f\t\t%s\n',ar1,datestr(rem(now,1), 'HH:MM
:SS'));
        end
        hold on; %(x);
        grid on;

```

```

        xlim([rem(now,1)-40.0926e-06 rem(now,1)+4.0926e-06]);
        abir='';c='';
        drawnow;
    end
catch e
    disp(e)
end
else
    set(hObject, 'String','Collect VH1');
    set(handles.Get_Data,'String','Collect');
    set(handles.node2Value,'String','Collect VH2');
end
guidata(hObject, handles);

% --- Executes on button press in node2Value.
function node2Value_Callback(hObject, eventdata, handles)
% hObject      handle to node2Value (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
if strcmp(get(hObject,'String'),'Collect VH2')
try
    x2=[];
    d2=[];
    set(handles.Get_Data, 'String','Collect');
    set(handles.node1Value, 'String','Collect VH1');
    set(hObject, 'String','Interrupt VH2');
    % while(t~=100) %Runs for 60 cycles - if you cant see the symbol, it
    is "less than" sign. so while (t less than 200)
    while strcmp(get(hObject,'String'),'Interrupt VH2')&&...
        strcmp(get(handles.Get_Data,'String'),'Collect')&&...
        strcmp(get(handles.node1Value,'String'),'Collect VH1')
        c2 = fscanf(handles.s); %reads the data from the serial port and stores
it to the matrix a
        opa2=length(c2);
        abir2='';
        for is=1:opa2
            if c2(is)=='z'
                if c2(is+1)=='3'
                    n=1;
                    for i=(is+2):opa2
                        if c2(i)~='e'
                            abir2(n)=c2(i);
                            n=n+1;
                        end
                    end
                end
            end
        end
    end
end
if isempty(abir2)==0
    d2=[d2 rem(now,1)];
    a2=str2num(abir2);
    % a2=a2';
    ar2=-3.86e-6*a2.^3+7.2e-3*a2.^2-4.41*a2+890;
    if ar2>80
        ar2=x2(length(x2));
    end
    x2=[x2 ar2];
    set(handles.axes1, 'XTick',rem(now,1)-40.0926e-06:...
8.0926e-06: rem(now,1)+4.0926e-06);
    set(gca, 'XTickLabel',datestr(rem(now,1)-40.0926e-06:...

```

```

        8.0926e-06: rem(now,1)+4.0926e-06, 'HH:MM:SS'));
        plot(d2,x2,'r');
        hold on; %(x);

fprintf(fopen('log2.txt','a+'),'%f\t\t%f\t\t%s\n',a2,ar2,datestr(rem(now,
1), 'HH:MM:SS'));
end
        hold on; %(x);
        grid on;
        xlim([rem(now,1)-40.0926e-06 rem(now,1)+4.0926e-06]);
        abir2='';c2='';
drawnow;
gg=0;
end
catch e
    disp(e)
end
else
    set(hObject, 'String','Collect VH2');
    set(handles.node1Value, 'String','Collect VH1');
    set(handles.Get_Data, 'String','Collect');
end
guidata(hObject, handles);

% --- Executes on button press in VH2_go.
function VH2_go_Callback(hObject, eventdata, handles)
% hObject    handle to VH2_go (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if strcmp(get(hObject,'String'),'VH2 GO') % currently disconnected
    try
        fprintf(handles.s,'z3200e');
        fprintf(handles.s,'z3200e');
        set(hObject, 'String','VH2 STOP')
    catch e
        errordlg(e.message);
    end
else
    fprintf(handles.s,'z3900e');
    fprintf(handles.s,'z3900e');
    set(hObject, 'String','VH2 GO')
end
guidata(hObject, handles);

% --- Executes on button press in VH1STOP.
function VH1STOP_Callback(hObject, eventdata, handles)
% hObject    handle to VH1STOP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if strcmp(get(hObject,'String'),'VH1 GO') % currently disconnected
    try
        fprintf(handles.s,'z2200e');
        fprintf(handles.s,'z2200e');
        set(hObject, 'String','VH1 STOP')
    catch e
        errordlg(e.message);
    end
else
    try

```

```
fprintf(handles.s, 'z2900e');  
fprintf(handles.s, 'z2900e');  
set(hObject, 'String', 'VH1 GO')  
catch e  
    errordlg(e.message);  
end  
end  
guidata(hObject, handles);
```

## **Confidentiality requirements**

This thesis is only for applying for degree of Master of Technical Sciences. Reproduction and use of thesis content is forbidden without the prior permission of the relevant author.