A. Demidov, A. Kuturov, A. Yudin, B. Krasnobryzhiy, C. Mikhail, and Rustam Borovik "Autonomous Mobile Robot Development in a Team, Summarizing Our Approaches" in Research and Education in Robotics - EUROBOT 2010, ser. Communications in Computer and Information Science, vol. 156. International Conference, Rapperswil-Jona, Switzerland, May 27–30, 2010, Revised Selected Papers: Springer Berlin Heidelberg, 2011. pp. 168-179.

Autonomous Mobile Robot Development in a Team, Summarizing Our Approaches

Andrey Demidov, Andrey Kuturov, Anton Yudin, Boris Krasnobryzhiy, Chistyakov Mikhail, and Rustam Borovik

Bauman Moscow State Technical University, IU4 Department,
2-nd Baumanskaya st., 5, 105005, Moscow, Russia
Lebedev Physical Institute of Russian Academy of Science, Division of Solid State Physics
Leninskiy prospekt, 53, 119991, Moscow, Russia
the-admax@yandex.ru, kuturov@yandex.ru, skycluster@gmail.com,
capitan_Flint038@mail.ru, mlazex@gmail.com,
borovikrustam@gmail.com
http://www.bearobot.org

Abstract. This article summarizes experiences of beArobot team members while developing mobile autonomous robots for Eurobot competitions. This text is an attempt to systemize approaches needed to successfully finish year-long project by a small team. Text covers method of formulating design tasks based on early robot strategy development from the plain text of competition rules. Further analysis of mechanical parts commonly used from competition to competition and unique mechanisms cover "how to make" and "where to get" questions. Basic principles of mounting electronic parts on mobile chassis and unified software platform approach for rapid modular program development finish the list of topics covered by the article.

Keywords: Eurobot, mobile robot, design, project management, good practices.

1 Introduction

21st century is no longer the time of individuals, it's the time of small teams of 4 to 7 members [1] able to solve any problems world is demanding including robot production and research. But how this is different from what was before, the impatient could ask? And the answer is in our educational system. Authors experienced engineering and fundamental university education but none included questions of team work.

You can never understand something until you have experienced it. This is true for humans and this is true for knowing what team actually is. Interested to get experience? One of the best solutions for engineers for now is to take part in team competitions like Eurobot [2]. Experience is guaranteed but what about its quality?

All teams are different like people but there are always traits that could be called basic. These traits allow some teams to win and others to lose. And there is nothing good or bad about that – as it has already been said we are only on the threshold of the age of new engineering education. Hopefully with time we'll see courses on team work included in university schedules. But for now we should accentuate two main elements that will be in the center of those courses – people and cooperation. These two define success of any project. People bring knowledge and this part is pretty well covered with present education. As for cooperation there is still a lot to be done but a positive trend is that this process can become a skill and there are plenty of such examples if you take football or any other team sport.

Bearobot is a team with 2 year Eurobot history already with all traits of a 2-year old child. No exact knowledge or winning strategies but some experience and wish to share. We hope this article will allow us to define first steps to methodology of team work and team cooperation basics first of all for ourselves and for those who's interested.

2 Start of project

Being a contest of teams presenting their mobile robotic solutions in a sporting spirit Eurobot is more than just colorful matches between two autonomous robots. It's a bridge to new level of education. Education including team work skills along with outstanding potential for practical knowledge and research. Competition on one hand and cooperation on the other both stimulate perfection.

When speaking of Eurobot project several parameters should be defined to understand what it is for participating teams. First parameter is time - project is limited in terms of time and usually it is given about 6 month to finish it. Second parameter is the number of team members. There could be different points of view on optimum number of people but our team believes that this number should also be limited at its maximum of 7 members. Actual number of people is changing with years but what proved to show tendency is the number of core forming participants - three is enough to finish the project. And another 1 to 4 are usually new comers.

When project is at its starting phase and team is gathered there should be a general plan template ready to organize its work. Such a plan reflects Eurobot specifics and evolutionary progress of the team while core members gain more and more experience and abilities to teach new comers and form engineering culture of development. Let's revise its main outlines.

For Russian students academic year starts in September. Teams usually would have about a month to search for new members until rules of Eurobot are published by early October. After that, project actually starts.

First step to make is obligatory elements and field construction for new members to feel the team and to start with simple and yet demanding tasks. Concurrent process includes brainwork on main robot strategy. Usually there will be 3 different tasks to be engineered with new mechanical solutions. On this stage strategy includes mainly analysis and estimation of task complexity. In future when it is needed to decide which task to actually realize an instrument of comparison is needed. This time we have to remember that competition goal is victory, and victory is based on points gained by the team. So naturally points will be the first parameter to analyze. But is it enough to form a strategy? Points usually would reflect complexity of the task determined somehow by the organizers. On the other hand one needs an independent parameter which would reflect real complexity specifically for his team. Such a parameter could be time to collect single playing element to score. When transferred to these 2 parameters all the tasks can be formally operated and thus we can define more suitable tasks for the team in an objective way. Let's call those 2 parameters a solver system.

To clarify first step estimations based mainly on member experience the actual development of mechanisms should be done. This step could be called prototyping as one has to "invent" a solution to a task and then test whether it works or not. When done properly this step brings accuracy to the solver system adjusting previous estimations.

Next step is actually another iteration of strategy forming based on accurate solver system. This time final decisions are made on which tasks to solve and which prototype to include in robot construction.

After all preparatory steps actual work starts. We suggest using distributed control system in the project. It allows true concurrent development of system parts by team members and forming reusable modules for later applications.

In case of modular design integration step is very important. Sometimes physical rework of parts may be needed but mainly this step forms true basis for final control realization and configuration. Ability of the team members to accurately and professionally organize robot control and module interaction determines success of the strategy and ideas put into system parts. Conventional wisdom even states that all possible strategies one could think of can bring victory to the team. Only strategy realization makes this statement come true or false in real life. That's why in the next section we'll discuss practical questions of robot building.

3 Growing-points

This section is called growing-points mainly because it describes all that knowledge gained by teams while progressing on Eurobot projects. Actually there can be defined 3 such points: mechanics, electronics and programming. Of course each point is bottomless in details and can be divided into less powerful aggregations with other names. But we are going to concentrate on general trends forming distinctive abilities and skills in competition participants.

This section provides information and personal beArobot team experience based on Eurobot 2010 rules [2]. For better understanding of further subsections it is advised to consult the rules. We believe this section to form basis for future research in our team and hope to attract interested individuals and teams to discussion on presented questions.

3.1 Mechanics

Generally from mechanics perspective robot for Eurobot competitions should combine common movement functions and a number of additional acting mechanisms. Eurobot rules change each year and this leads to changes mostly in acting or in other words manipulating schemes. Chassis and navigation system once correctly implemented do not need any significant changes. This fact brings modularity from the very start of another project for a team. For mechanics development it should be common to whenever possible to produce parts able to be transferred to other robots of the team as well.

Robot easy to manufacture is rather difficult to design. Ideal robot should consist only of standard parts, available within different production sectors like automobile, furniture, modelling sports, etc. Such a robot will possess a number of advantages:

- Easy production – accurate and detailed design of mechanism part is not needed, as commercially available part is usually optimal and reliable in design;
- Reliability – when constructed with reliable parts, quality of which is proven by years of production;
- Simple assemblage – mainly because of completeness of parts being used, freeing from part's mechanics workout;
- Repeatability of assembly – using unified parts makes it easy to get to modular design, which will allow easy repeating of modules.

Practically it is almost impossible to design ideal robot. Often robot would only partially consist of standard elements. If we look at the means of realization of any mechanism for Eurobot competitions we could define the following list of possible options in order of decreasing preference:

- Usage of commercially available parts;
- Self-design parts and order professional production;
- Self-design parts and further self-realization.

The quality of resulting robotic solution is fully dependant on experience, skill and knowledge of mechanic in a team. Not to start all over again with each new project it is good to track successful solutions whether they are a product of a team itself or whether they are a product of team's interaction with other teams on competitions and other activities. Understanding the impact of mechanical solutions on the whole robotic project, analysis of mechanisms commonly used on Eurobot robots seems interesting, challenging and useful idea we hope to realize in our future work.

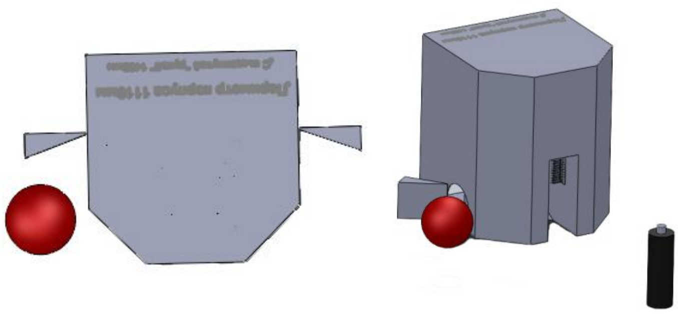

Fig. 1. Robot for Eurobot 2010. General view.

Let's analyze fundamental principles for constructing robot mechanics by the example of Eurobot 2010 rules. It is to be said that from the very start we decided to concentrate only on corn collection. We believe it is not necessary to review and analyze each and every part on the robot but for a single mechanism of collecting and storing corn inside the robot in this case.

For general concept of the robot let's refer to figure 1. Left side of it demonstrates robot from top view, and right side – axonometric projection of robot body. In the figure you can also see several playing elements included for size reference purposes.

Fugure 2 complements more information on how robot systems are designed and reveal general ideas. Right side of the figure presents more general front view and left explains how inner mechanisms are organized. You can see numbers in the pictures. These numbers are to help locate parts in our further system explanation.

Main idea to manipulate corn elements consists of several parts. First part is under number 4 in the next figure. It's a sort of brush which main purpose is to bring corn down. Fake corn is able to come through robot interior and thus eliminating the need to avoid it while cruising in search of playing elements. Number 1 represents sensor array that allows detection of corn. Whether it is fake or real corn element – control system is able to make distinction with original sensor placement. Number 3 represents grabbing system of the robot. When real corn is detected elevator is activated grabbing corn and transporting it to the top of robot body to release in one of two storage containers. Finally number 2 represents supplementary ability to push balls towards the scoring area.

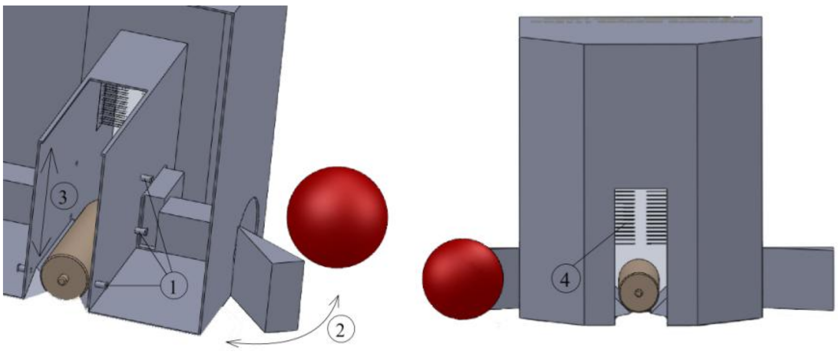

Fig. 2. Robot for Eurobot 2010. Collecting mechanisms basics.

Let's review corn collecting mechanism. Its purpose is to grab lying corn from the field and to get it to the storage containers onboard. One of the important tasks of the mechanism is precise positioning of corn as requested by storage and unloading mechanics. Let's define corn for processing as lying in the open-ended passage (which is wider than corn diameter for 15cm).

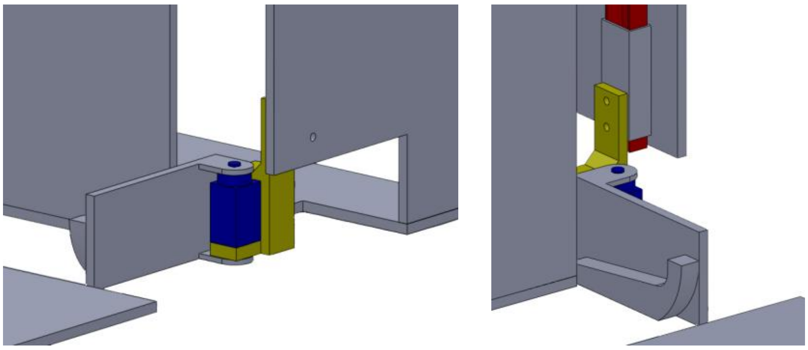

Fig. 3. Corn collecting mechanism. Door for precise corn positioning.

Main elements of the grabbing mechanism are two doors able to open and close with the help of servos. On figure 3 you can see closed state of one of two doors needed for positioning. When corn is detected by sensors doors are brought to closed state. While closing, doors can slightly move corn element thus positioning it between two doors in a known state. Door and servo are fixed to guidance track allowing both to travel vertically. Vertical movement is due to flexible wire system activated by motor. Let's analyze origins of parts in described system.

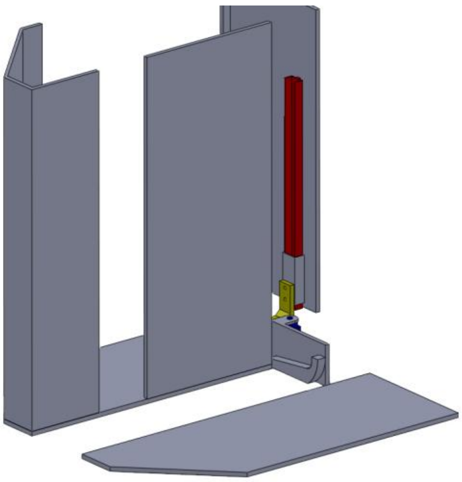

Fig. 4. Corn collecting mechanism. Door for precise positioning.

Servos and motors were bought as they are freely available on the market. Some effort was applied to find suitable track and wire, but they are also bought elements.

As doors are original in design they had to be produced by the team. Main material to process with laser cut machine was PVC. Rollers for wire guidance were produced with 3D printing machine. As one can see while constructing grabbing mechanism we used commercial and self-made parts, produced with modern machines kindly provided by the Russian Eurobot NOC [3].

While system construction we discovered several specific mechanical features to be highlighted. Suitable guidance track was hard to find. A good choice appeared to be a part from writing table's drawer move system. Several guides from different manufacturers were tested and best was chosen for our purposes. Wire for vertical movement of the carriage was chosen from fishing equipment. Special flexible and strong fishing line was used for that purpose.

Rollers satisfying our needs for compact design could not be found and thus decision was made to construct them individually.

Described examples show our suggestion to use commercially available parts, limitation factors being search time and availability of suitable designs. Depending on both factors self-development may be required.

3.2 Electronics

Electronics and electronic parts have a long history. If we look back at those rates of growth in 20th century, according to so called Moore's law [4], which are still true for the whole field, we'll see a never ending trend of evolution of electronics.

Some advantages of such development include simplification of usage of different parts which some time ago were considered too complicated to be used in ordinary design. As a consequence we have a large amount of electronic devices to choose from these days.

If we take electronic design in terms of Eurobot competitions we'll see correlation to previous statement and to the large choice. Naturally each team would have its own thinking of what is best for their current robotic solution and thus one can see a lot of differently organized electronics onboard of robots. We are not going to advance in the question of what is the best and the only good practice to arrange electronics for now, but we are going to discuss our way of doing it and ideas lying beneath.

When you get in touch with Eurobot for the first time you see competitions. From spectator's point of view the reason to come is to see challenging opposition and colorful matches. For him it is natural to expect each team's aim to be victory. But then if we take a look at the same question of participation from competitor's perspective we'll see that it is more of education in place. We'll see Eurobot as an educational project.

Now if we turn back to electronics taking into account educational perspective we would suggest among other variants of buying electronic parts from any third party to design it within Eurobot project as other mechanical or software parts.

Now what about difficulty level? Robotics has immense potential for invention and thus each solution can radically be different from others. Depending on the problem to be solved electronic solutions will be of different difficulty. The good point is that these solutions can be simple and thus designable by the new comer from the very start.

For our team it is usual to use 8bit microcontrollers of AVR-core nature. Having one microcontroller with 32kB of flash is enough to organize simple navigation and sensing needed to avoid obstacles. As a matter of fact our experience shows that that

A. Demidov, A. Kuturov, A. Yudin, B. Krasnobryzhiy, C. Mikhail, and Rustam Borovik “Autonomous Mobile Robot Development in a Team, Summarizing Our Approaches” in Research and Education in Robotics - EUROBOT 2010, ser. Communications in Computer and Information Science, vol. 156. International Conference, Rapperswil-Jona, Switzerland, May 27–30, 2010, Revised Selected Papers: Springer Berlin Heidelberg, 2011. pp. 168-179.

is the minimum system requirement to be able to take part in autonomous robot contest.

To be able to progress with time it is good to save previous solutions and use them in new projects. This way any team can become successful in several years if there is no source to learn from other than bare competition and preparation experience. For that idea to work it would be necessary for electronics to provide a means to independent developing of devices and their easy interconnection in the future. Each device can be constructed by different member of a team or at different times. A good practice in such a case would be sole bus selection made from the very beginning. Our team prefers CAN bus for that purpose.

With time growing experience for team members will bring more interesting ideas to electronic organization of course. Some key aspects of choosing approach to electronics architecture design are: work organization of team members, skills needed to work with the system, maximum abilities and functionality of the system, scalability.

In our case we prefer modular approach to electronic design. This allows making common parts identical and to design or redesign only those parts which cannot provide means needed for current project. Such general electronic part is shown in figure 5.

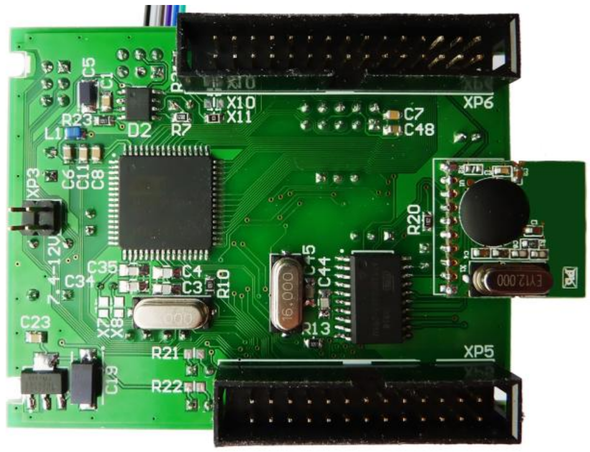

Fig. 5. Microcontroller embedded board used for control on the robot.

Question that is closely connected to bus or, more generally, to architecture selection but is more practical is the number of wires in the system. It is obvious that the less wires are on the robot the better. Large number of wires means more work while mounting and even more work while debugging if something goes wrong. Sometimes this would lead to different influences of noise effects preventing the system from correct operation. How to make the number of wires small? First measure is to introduce differential buses like CAN physically consisting of a definite number of wires. And second measure is to put separated microcontrollers as close to their field of action as possible connecting them with a bus for communication.

Finally if we touched electronic mounting onboard of mobile robot let's research how to do it right. Usually questions of electrical noise effects on a mobile device are not trivial. Such effects have several origins one of which is appearance of static electricity. As such effects could influence normal operation of the whole system or even damage parts, we should name a number of rules to be fulfilled while designing electronics mounting scheme.

Problems of static electricity arise when designing aircrafts (moving airflows different for parts of the fuselage cause difference in electric potential) or when designing powerful antennas (formed by antenna electric field could make parts of house interior also different in electric potential). Special and yet simple measures to discharge the body of a trolleybus to ground can be easily observed on the road. You can see now that such questions are rather common for a range of devices. Let's investigate more on what causes such problems on mobile robot.

One of the causes of static charge on the robot body experienced when touched by hand could be triboelectric effect [5]. Most popular materials used in Eurobot robot design are aluminum, different plastics, acrylic glass and wood for the body, rubber and polyurethane for wheels, acrylic paints are used for main playing field, and human skin as from time to time contact. All these materials have different electrochemical potential. The row from most positively charged to most negatively charged will be: human skin, +, aluminum, 0, wood, -, acrylic, rubber, -, polyurethane, plastic ("+" or "-" indicate large gap of potential between elements). Materials brought into contact will attain different charge – positive for those closer to the start of the list, and negative for those closer to the end.

Usually when robot is placed on the playing field one can expect acrylic paint to cause electrostatic effect when in contact with polyurethane wheels. This effect is much less strong if wheel material is rubber. While moving on the field robot tends to collect charge if there is no special means introduced to its design.

Let's review what the consequences of such physical behavior are. If this question is left unsolved by the designer it can cause sparks and unpleasant sensation while touching the robot with hand. Another more technical aspect includes ability to cause damage to microelectronic parts on occasional discharge. This leads to the need of some sort of protective package for most sensitive parts and thus to more work. And finally excessive charge forms electric fields able to cause noise effects of different sort in electronic schemes. This can lead to unpredictable and difficult to track and debug unstable functioning.

First solution is to keep charge separated locally in place of contact. This solution is simple and pretty straight forward. Its main idea is closely connected to the physics of triboelectric effect and states that when charges on both surfaces equalize each other there is no more charge gathering in place. In this case we have to put dielectric as close to the contact zone as possible to prevent electrons from travelling along metals used in design.

Second solution is to transfer gathered charge to onboard battery along metallic elements of the robot body. When most of the body is made from metal it is easier to connect main battery "–" electrode to it. This way we let electrons to flow easily to and from energy source. As long as we use polyurethane wheels electrons will be generated on their surface. This will help "-" electrode of the battery and will not waste its energy on compensation of supplied potential. On the other hand if contact zone is presented with metallic roller it will tend to gain positive charge and thus will waste some of the battery's power. Both effects seem to be negligible but need to be researched in future work.

Finally it is to be said that a good practice is to ground the body of a robot, connecting metallic parts together and to the ground of the battery. This measure allows more predictable system operation as noise generated by the robot itself with its electronics and static charge gained while moving is neutralized. More strictly it provides noise immunity to control systems of the robot. Additionally this measure allows neutralization of sparks and harmful discharges between robot and its environment.

3.3 Programming

Currently, the hardware platform of our robot consists of similar modules based on AVR MCUs, namely AtMega128. Each of them serves particular peripheral devices (e.g. sensors or motor drivers) and is connected to common bus used for data exchange. The conditions of work are rather harsh in aspect of various noise and limitations (i.e. time and energy limits). According to this further on we suggest a concept of robot control system.

The logic of robot is built via "atomic" blocks – functions. They form robot's behavior by connections between themselves. Actually robot consists of electronic modules, each of them executes a number of functions, e.g. controls motors, processes signal from sensors or prevents collision with obstacles.

These functions can be randomly connected to each other according to neuron-network construction principles: output signal of a "neuron" can be transmitted to several destinations (one-to-many).

Linkage between functions is organized at system setup and in some cases when already in use. The best configuration for functional blocks satisfies all stated network requirements.

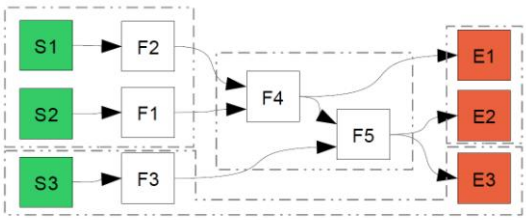

Fig. 6. Possible outline of control system. Sensors (S1-S3), data processors (F1-F5), executors (E1-E3). Single hardware modules are marked with dotted line. Arrows show data flows.

A platform is needed that will make it possible to implement such structure of robot control. Actually it is implemented as shown on figure 7.

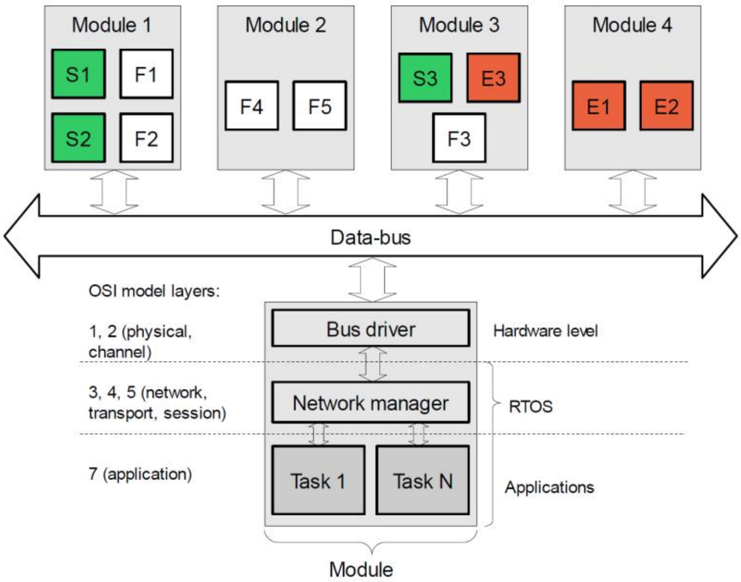

Fig. 7. Control system structure.

It is necessary to separate actual network structure from its logical representation. Implementation means installing RTOS (real-time operating system) with multitasking support on MCUs that provides generic and platform-independent programming interfaces to hardware, network and MCU's peripherals.

Such OS (operating system) should be:

- Flexible (because we want to use the same operating system on every controller despite their functionality destinations);
- Tiny (considering the fact that applications need a lot of resources, especially memory);
- Functional enough to provide any needed routines at different levels of abstraction.

Finding such an OS frees us from writing any trivial code of our own and greatly increases time for essential tasks. Let's take a look at some of such systems, namely Nut/OS [6] and FreeRTOS [7] and choose the best for our purpose.

First of all, both OS support AVR MCU family and its capabilities: timers, interrupts, IO-ports, internal and external memory and other functionality. Also they have API for multitasking (managing tasks, inter-task communication routines, etc.). But Nut/OS as opposed to FreeRTOS has also memory management routines, better bus support (CAN, UART, ISP, TWI...), device support (ADCs and DACs, timers and other peripherals), network protocols (IP, TCP, PPP, ICMP...) and filesystem support (FAT12, FAT16, PHAT). Besides it provides configuration parser that produces makefiles and header files with different options for target OS's kernel and modules.

Within all advantages and power of Nut/OS there is expectable redundancy in functions it provides from our perspective of robot control. In particular we believe file system means of presenting every possible actor (e.g. devices, data, and processes) as a file are not crucial for automatically and autonomously controlled systems, actually this approach introduces a level of abstraction we could ignore in a definite system, but sometime later they could come in handy of course.

For now we will ignore most of excessive functions provided to carry on with research of existing real-time systems, their abilities and appropriateness for robotic tasks. In our opinion both FreeRTOS and Nut/OS are documented and coded well, what makes them a good start. You can even find comprehensive guide to FreeRTOS's internals on its website that we found useful for studying advanced programming and multitasking basics.

At last we came to conclusion that it's better to try Nut/OS, due to its rich set of drivers and interfaces. Of course if it proves to be suitable enough we still need to introduce a lot of modifications to it to actually serve our purposes. Hopefully we believe to discuss our experience in the next articles.

4 Conclusions

This article summarizes experience and understanding of Eurobot competitions gained by still young beArobot team. All presented information is fruit of labor of many current members of the team but of course the influence of those once in a team will remain within team spirit and culture. With this article we hope to start a series of arrangements to bring more order to Eurobot projects at least for Russian teams.

We slightly outlined future research topics and points for professional growth. Besides well known topics like mechanics, electronics and programming we believe team work skills to be crucial for success of any Eurobot project and Eurobot movement in general. Some practical questions and examples were presented in the article to explain our current project and those methods and approaches we believe to be good practices in Eurobot design.

References

1. Belbin, M.: Management teams: Why They Succeed or Fail. 3rd ed., Butterworth Heinemann (2010)
2. Eurobot, international robotics contest, http://www.eurobot.org
3. Eurobot National Organizing Committee (NOC) of Russia, http://www.eurobot-russia.ru
4. Wikipedia article, http://en.wikipedia.org/wiki/Moore's_law
5. Jonassen, N.: Electrostatics. 2nd ed., The Springer International Series in Engineering and Computer Science, v.700 (2002)
6. Nut/OS real-time operating system, http://www.ethernut.de/en/firmware/nutos.html
7. FreeRTOS real-time operating system, http://www.freertos.org