Distributed Control System in Mobile Robot Application: General Approach, Realization and Usage

Andrey Vlasov, Anton Yudin

Bauman Moscow State Technical University, IU4 Department,
2-nd Baumanskaya st., 5, 105005, Moscow, Russia
vlasov@iu4.ru, skycluster@gmail.com
http://www.bearobot.org

Abstract. This article is aimed to give a general idea of how to organize robot control with a distributed network of embedded devices. It is an attempt to formulate simple concepts that will allow easier cooperation for team members during development. Suggested approach when properly described, understood and implemented is expected to form a good platform for concurrent individual software and hardware development and for educating new team members. Giving flexibility such an approach leads to convenient modular robot design. Realization of proposed approach is also introduced with a hardware version of embedded controller board, being the "heart" and the "brain" of any module on the robot.

Keywords: Eurobot, mobile robot, distributed control system, modular systems, team development.

1 Introduction

It's common to hear from different people now and then the same question about new technologies expressing fear or lack of trust in evolutionary scientific process. There's hardly a thing we can tell them to reassure that things are going their natural way but to name the reason we are doing this global research for. And the reason can be found deep inside any of the living creatures' mind. Being an open living system brings a strong urge to put order in the chaos of life. Order can be thought of in terms of construction as an engineering process – the only process to check whether our knowledge and thinking are correct and fully reflect nature of things. In other words we cannot predict anything until we have experienced it, built it.

"Who are we and what are we doing here?" – is also a question that rises every now and then in us. The true nature of such questions can hardly be known at our current state of understanding. But as many other facts which we cannot explain they exist. And the only way to find answers is to search the outer world and to get to know ourselves better step by step. Researching and building robotic systems could be one of such ways as they are similar to living beings in many aspects of existence.

Speaking of artificial systems we could turn to Herbert Simon [1]. In his book he explains how artificial becomes a part of natural. And more precisely he explains and

demonstrates that human thinking includes natural and artificial formations. There are no motives not to believe this and many other scientists. And if we take these ideas as true we could reveal many mysteries in ourselves by constructing more and more intelligent machines. At least the artificial part of our own self could be finally understood.

Hopefully ideas briefly expressed above will help to explain the reason of the research on suggested topic. Throughout the history of science many efforts were made to make Nature clearer. The work partly described in this article is a small contribution to this movement. Concentrated mainly on control processes and communication its final destination is to describe a system similar to human nervous control system, highlighting important mechanisms forming intelligent behavior.

It should be noted that this article indicates an early stage of research and thus is also aimed at finding interested opponents and colleagues to discuss and evolve the suggested approach.

2   Why distributed?

Luckily having participated to several robot projects for Eurobot competitions the author experienced several ways of organizing control on a mobile autonomous system. Each project was different in a center topic of development. Approaches ranged from a complicated control system using a compact PC with its power to calculate and style of higher level programming to a simple embedded microcontroller as an attempt to find more simple ways to organize the whole robotic system. All control systems realized no matter how complex they were contained serial approach to data processing and reaction, meaning that all decisions had to be done by one and only one processing unit. And it worked. So why does distributed control appear?

Before we even start looking at the arguments it would be useful to define what author means with "distributed". Distributed in our case refers to physical structure of control system represented by separated processors which could work as a whole coordinating each other and cooperating. The most comprehensive system of such kind is human brain and the most known human constructed system of such kind is World Wide Web.

2.1   Team cooperation and education

The first and the most influential reason grows from competition characteristics and here consulting the competition site [2] for the rules and general idea is suggested to help understand the argumentation.

Being a part of a team revealed a lot of trouble young engineers have when for the first time they get in touch with a not so simple integrated system design. Less experienced students simply don't know how to start and thus need a strong supervisor to organize their work for them, more experienced students show tendency of doing everything alone. The general idea that is provided by Eurobot is for participants to experience team work and cooperation. None of the above natural

tendencies, demonstrated by the students could possibly lead to fully comply with this brilliant idea in a half-year project. At least author's experience doesn't include positive results of such behavior.

The idea that could possibly save the situation lies, as usual, in between. For those who start over we have to provide a general plan, that will work but still give them enough space to make their own decisions. The plan should be supported with general equipment that could be expanded by the team members to current project needs. And decision making should allow mistakes to learn from. If we take a half-year period and the real amount of work to be done to successfully finish the project it becomes obvious that the whole scheme with freedom should be well thought of.

Suggested distributed approach steps towards the scheme which will help new comers understand the value of team work and to construct a working solution to the rules of the competitions on a team basis.

Speaking of teams from 4 to 7 people Eurobot provides at least 7 subsystems[1] to be realized. That means that each team member could possibly design at least one subsystem. It would be natural to make the process of design and construction concurrent where possible to benefit from team work though parts of final integrated design should be of course coordinated from the very start. High quality coordination requirements lead to general methodology of educational and commercial projects, which can be revised and improved each competition year.

Summarizing statements above, we come to a competition solution as defining separate work-parts of robot and giving those parts to team members for concurrent design. These parts include mechanics, electronics and programming. Design process of work-parts should be as independent from other parts as possible. By carrying out robotic project this way we benefit from team work and individual experience of team members. And in this case for a wide range of purposes (new comers, education, rapid design etc.) we need common hardware and software core which allows broad extension to current project needs on the educational level of the team member.

2.2 Effective system resources utilization

The question of natural resources preservation is becoming more and more actual as years pass and planet reserves are exhausting. Problems of human-designed system effectiveness are closely connected to this question.

Main resource that is consumed onboard of any mobile robot today is electric power. Generally all mobile robots use batteries of different kinds to work autonomously. As battery capacity is usually perceptibly limited question of power consumption has to be accurately considered when designing the system. As a consequence this leads to overall better effectiveness in terms of power consumption per action when compared to stationary power supplied systems.

To understand the order of consumption difference we'll compare an average EPIC standard embedded PC board with a Celeron M processor at 600MHz as the most

---

[1] This article is not aimed at precise system design description and thus the number given is mainly for reference purposes. Still the number was taken from a real project of Eurobot 2010 carried out by beArobot team.

powerful control solution, an average PCI104 standard embedded PC board with Geode LX 800 processor at 500MHz and PC104 standard embedded PC board with Vortex86DX processor at 800MHz as an intermediate control solution, ARM-core based microcontroller board at 36MHz as an advanced distributed control solution, and AVR-core based microcontroller board at 16MHz as a simple distributed control solution. The next table gathers all information needed for analysis at one place. Again this analysis is mostly approximation of the real system, showing only order of difference between radical solutions. Time lasting estimation is done using typical battery capacity of 2100mAh and with assumption that processing unit is working alone on same battery.

Table 1. Data processing unit's power consumption comparison in mobile robot application.

Architecture	Manufacturer info	Power	Lasting time
Celeron M @ 600MHz	12V x 1.7A	20.4W	1h15m
Geode LX 800 @ 500MHz	5V x 1.3A	6.5W	3h50m
Vortex86DX @ 800MHz	5V x 0.74A	3.7W	6h50m
ARM-core proc @ 36MHz	11.1V x 0.16A	1.776W	14h10m
AVR-core proc @ 16MHz	11.1V x 0.09A	0.999W	25h15m

As you can see the better the calculation speed is the more power is consumed. It is to be noted that large power consumption for PC-compatible processing units would usually be spent less effectively because of different types of software latencies. Moreover if application is demanding real-time processing as most of robotic applications do software latencies are becoming a crucial point in system design. As microcontroller software is simpler and "closer" to hardware it is much easier to realize a real-time application with it.

If we do some search on distributed systems research by other people we can find useful confirmation [3] of above assumptions: "Except for simple microcontrollers with low performance and only basic features, modern processors exhibit serious drawbacks when employed in embedded real-time applications. …No high performance processor enables the exact calculation of a program's execution time even at the machine code level, as is possible for the 8-bit microprocessors." That's why any PC-compatible board must be accompanied by at least one microcontroller in a robotics application.

Complicated tasks in a robotic application that will require high performance processor will be: vision – image processing; and tactics[2] – large amount of data processing. These tasks unlike others (like navigation or even more strict sensor calculation) do not demand strict real-time and thus can be realized on a general high performance PC. But again if we look at power consumption of such PCs like Celeron M we can see that equivalent network of microcontrollers could count from 20 pieces of solely AVR-core based (equivalent frequency 320MHz) to 11 pieces of solely ARM-core based (equivalent frequency 396MHz).

---

[2] Strategy questions could be even more demanding, but we are staying in Eurobot competition limits, assuming strategy is formed by people and programmed statically. Tactics in this case can be thought of as a number of current preprogrammed actions put together dynamically by the robot.

A. Vlasov and A. Yudin "Distributed Control System in Mobile Robot Application: General Approach, Realization and Usage" 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. 180-192.

If we take the most consuming choice as a reference for the whole system top allowable power consumption for data processing purposes we could build two possible types of distributed systems instead:

- Main layer consisting of AVR- or ARM-core microcontrollers and one dedicated Geode LX 800 or Vortex86DX compliant processor for high demanding, complicated tasks;
- Network of devices based only on AVR- or ARM-core microcontrollers.

Both types are able to solve all tasks described but offer better scalability and power consumption quantization. Both leading to overall better system effectiveness and better correspondence to a definite real-time system task. Naturally compared to a centralized serial processing this approach will lead to equal or better performance and to longer battery life in general.

2.3 Cost effectiveness

Questions of costs usually arise when project resources are limited which is true for all student Eurobot projects the author was involved in. First thing to mention here is that robot project no matter how simple the final solution will be demands a lot of resources which could be usually converted into money costs or time-to-produce equivalent for general model presentation.

Resources needed to build a robot include mechanical parts, electronics of different sort and software depending on chosen electronics. When designing a new robot each team has opportunities to do everything by itself or partially order parts from third parties. Question whether to use third party parts or not is driven by quality characteristics (whether team members can achieve needed by the project level of production quality) or by time characteristics (whether team members can do all work needed for the project in time). And of course project's budget should also be considered when deciding on such questions.

The aim of this article is not to propose a detailed model of costs in a robotic project, so we are going to analyze only the influence of different control system solutions based on the previous subsection. And again, this analysis is mostly approximation of the real system, showing only order of difference between radical solutions.

Table 2. Data processing unit's cost for mobile robot project[3].

Architecture	Equivalent relative unit cost
Celeron M @ 600MHz	7.7
Geode LX 800 @ 500MHz	5.7
Vortex86DX @ 800MHz	5.7
ARM-core proc @ 36MHz	1.3
AVR-core proc @ 16MHz	1

---

[3] For obvious reasons we cannot discuss current absolute costs. Author believes relative costs to be more accurate with time and containing more information. Base values for presented costs are actual for February 2010, Russia, Moscow.

Above table speaks for itself – simple AVR- or ARM-core solutions proposed for distributed control system design are much cheaper than other options. Other than that they can be designed and produced by the team itself. This is not true for other options which are much more complicated in design and cannot be realized by the team in Eurobot project time period.

Another point concerns reliability and parts replacement in case of malfunction. It is a usual practice to have spares for robot's important parts. In case of a single Celeron M solution we'll have to pay a lot more to have a part which we might not even use if all goes well. Almost the same situation is happening if we use Geode LX 800 or Vortex86DX in the design.

Positive outcome of a distributed system consisting of simple identical parts allows us not to have spares for all of them and thus reduce costs for control system. Quantization of total cost in this case allows us to assemble control system closely related to project budget. Possibility of self-development of distributed parts leads to engineer experience for team members and even more reduction in total costs.

2.4 General software and hardware simplification

In suggested distributed system we can put less effort in complexity of separate system parts. Most picturesque analogue is human brain with single neuron as a basic part of control system. It is believed that we know the exact mechanism of neuron functioning [4]. Simplicity of each neuron is determined by its activation function, hiding system complexity in numbers of neurons and their interconnections.

Though full brain function yet cannot be reproduced in artificial design we can benefit from learning how Nature organizes complex systems. According to H. Simon [1] each complex system is hierarchical in structure. Moreover because of such hierarchical structures we are able to learn how world is organized as all of hierarchies possess similar properties independent of current realization. To reveal these similarities Simon suggests paying more attention to intensity of interaction between system parts rather than any other property (like spatial location of parts for example). Hierarchy levels are actually intermediate stable forms of systems on their evolution way. And speaking of interaction it can be observed inside a level between its parts and also between levels which is the most interesting part for a researcher.

Such argumentation could strengthen and help explain the idea of simplifying system parts and arranging them in interconnected nets as it reflects Natural tendencies. Simplicity will influence individual developers on the team giving them more freedom in work. To achieve that, special layer of software should be added to the system to glue all parts of the final project together. This layer has to organize communication inside the whole configuration and is the most influential part of it. That is why it has to be developed with much care making it the most difficult part of system software on the robot. But once realized it can be reused in different projects. Moreover it can become a part of educational methodology for young developers restricting their style of programming.

If there is a question of simplification, education and general usage involved we should pay more attention to how the system is realized. In such cases we should think even more than usually of such a design that will help its user in his work.

Distributed system we are discussing here contains software and hardware elements. The user being interested in solving a number of tasks put for him by the competition rules should have easy and clear access to system functionality. Speaking the words of D. Norman [5] a good design should comply with two fundamental principles:

- Clear conceptual model;
- Visualization.

Using these principles and other design techniques is important on all stages of development. Design defines how easy it is to work with the system and thus success of the principle, current project and future projects which might be based on current one.

Summarizing main ideas of subsection it is important to highlight possibility of simplifying a system design by introducing separation to control parts as it is done in advanced Natural control formations like human brain. By making each part of the system simple we develop a hierarchical system structure with two main layers. One consisting of parts themselves and the other consisting of interconnections and communication means. And if the first one is represented mostly by hardware parts, then the second demands accurate universal software core. To be successful both levels must be designed easy to use.

2.5 Robust design

When we speak about mobile objects like cars or robots we think of highly reliable devices able to operate even in harsh environments. Usually to achieve such reliability parameters of systems are highly reserved. In case of electronic control system it is a good practice to introduce duplication into design configuration. Robust in this case would mean able to work when some random parts of the system are not working right.

First question of robustness reflects its essence. There should be a mechanism in a system to continue operation while some parts show malfunction. Solution to this question lies on hardware and software levels. On hardware level there should be a physical way to reallocate control lines from a broken part to a working part. On software level there should be a mechanism to determine which part is broken if broken at all and reallocate its control functions to other working parts.

It is known that human brain is highly robust and mechanisms used for communication between neurons allow quick reorganization for alternate routes. When operating as usual these mechanisms allow load balancing and in case of emergency they ensure save of functions. Artificial robust control system should possess similar properties, the most important question being physical alternate interconnection route organization techniques.

Second question of robustness is how quickly system can be repaired in case of malfunction of some or all parts. This question is highly connected to the one discussed earlier about spare parts.

Distributed system naturally will have more potential to develop robust techniques rather than highly concentrated system. Mainly because questions of robustness are similar to those questions vital for distributed system's existence.

2.6 Research platform

Inventions of the past allow us to progress to better solutions of the future. Ability to discover does not belong exclusively to fundamental sciences. If we refer to H. Simon we'll see that he considered design engineering a special branch of activity different from natural or human sciences and different first of all in methods of progressing. In his book [1] Simon takes a step to formulate what should sciences of artificial teach and what to include. And once over with his book it becomes obvious that sciences of artificial or design engineering mainly use synthesis rather than analysis. All tools and methods supplied to young engineers in universities should mainly concentrate on design techniques and they require cardinally different, special logic. Before Simon science was seen from such a perspective that artificial was a product of science and there is nothing to investigate, to research in it. On his behalf Simon highlighted the meaning and presence of artificial in Nature.

Research in engineering means a special strategy of search by synthesis. It's much of a trial and error strategy. And it can be explained again with brain analogy. When we cannot analytically come to understanding how Natural system works we have nothing to do but to try reproducing it by synthesis of known elements. We know how neuron works but we don't know how a net of neurons gains ability to intelligent behavior. We could try to engineer a system with similar behavior and discover yet unknown correlations which could bring us to a new level of understanding of how Natural process is organized.

Distributed system being described could become such a research platform. Distributed parts staying physically unchanged could serve as an inexhaustible source for software development and interconnection organization.

In spite of the wish to simplify system parts it would be right to address another scientist, John von Neumann. In his unfinished work [6] von Neumann searched answers for two main questions which could possibly be asked later in our research work:
- How to construct a reliable system out of unreliable parts?
- What logic organization should be sufficient for a system to reproduce itself?

John von Neumann believed that when you start to work in a new scientific field it is right to start with problems that could be clearly formulated, even if they refer to ordinary effects and lead to good known results. Later a strict theory explaining such results could become the foundation of progress.

Proposed distributed system could possibly carry no unknown elements in terms of nowadays' electronics or software engineering but it carries the potential to discover.

3 System structure, principles and realization

To start describing how distributed system is meant to be organized let's briefly summarize background and ideas that actually influence the future configuration (you can find more accurate argumentation in the previous section). These points describe goals to realize in proposed system and consequences of such realizations:

- Simplification of team work – method to organize cooperation of developers;
- Simplification of system hardware – capabilities to self-development of system hardware and education capabilities;
- User friendly design – more system users, general design success;
- Introduction of basic control modules – modular design method, capabilities of rapid development;
- Cheapening system elements – more accurate quantization of system costs;
- Effective resource utilization – more accurate quantization of system power and performance;
- Foundation of future research platform – base unchanged hardware allowing easier and quicker research in interconnection and communication techniques, robustness and software control.

Now we know exactly what we want to achieve with the new system and it's time to discuss the means and methods. In other words we must understand how those goals can be accomplished. For now it should be clear that we are discussing a control system mostly to be realized in hardware and this system has to be distributed in its core organization.

Main system structure and at the same time usage diagram of distributed architecture is presented in figure 1. Schematics is meant to be rather straight forward to understand most of ideas from the figure itself but there are several important remarks to make.

It would be good to have a universal control system at the end of development as success of any design is determined by the number of uses and users. But it should be noted that current version is developed in spite of Eurobot competitions and thus reflect the structure of challenges typical for competition rules. This, of course, doesn't necessarily mean that the system cannot be used in other applications; it means that some parts of the system should be considered as universal and others as application specific.

If we take a look at figure 1 we'll see that there are 7 embedded control boards each on a specific system level. Levels should be considered universal and thus present current distributed control system architecture understanding. Any mobile robot control application is meant to be same in level structure.

On the other hand total number of embedded boards and their number on each level are considered to be application specific. Generally the number of boards is equal to the number of system tasks.

If we take Eurobot match as an example we'll see that first of all robot must be able to move ("Navigation acting level") – this time we use a simple differential drive for that purpose requiring only one control unit as there is not much data from sensors to be processed.

Secondly there are always a number of mechanics oriented tasks to realize. For example: robot should gather balls to some storage inside its body and unload them when needed. Embedded boards to realize control for such tasks are located on "General task level". And this time there are 3 tasks, each with unique mechanical background, sensors and processing, and 3 control boards for them.
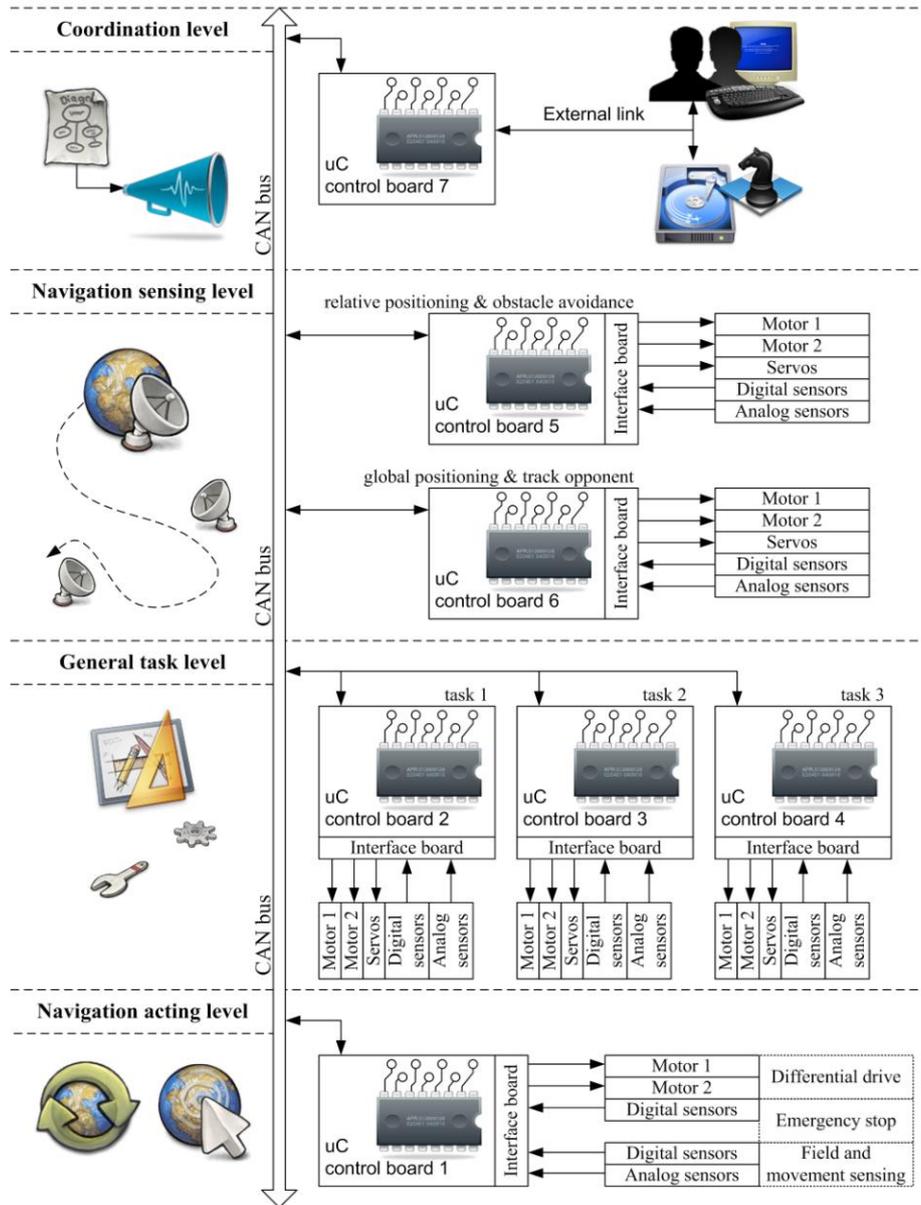
Fig. 1. Structure of mobile robot distributed control system.

Thirdly there are questions of navigation which refer to localization of robot itself and objects in its environment. Questions from one side include relative positioning techniques which allow gathering information relative to current robot position, tracking robot movement and presence of stationary obstacles on the way. On the

other side, accurate localization of robot and its opponent in the environment. These tasks are represented with 2 control boards in the model structure.

At last there is a level that represents the whole distributed system – "Coordination level". Control boards belonging to it contain software core needed to start the system, possibly support its functioning while system is active, they provide system with additional calculation power and memory to share and finally they form a gateway to out-of-system control and communication. If you lack resources for some task this is the first place to consider additional board modules. Current system architecture is similar to asymmetrical multiprocessor architecture reviewed by M. Colnarič and others [3].

When designing a control system of mobile robot there is a lot of debug work for developers especially in programming field. Taking into account possible amount of embedded devices normally we should carefully choose their placement onboard the robot. Decision depends on easy access for debug and embedded program uploading on one hand and position close to controlled hardware for less noise emission and reception on the other.

Figure 2 helps to understand how hardware installation and reprogramming is simplified by putting all hardware descriptions in a single configuration profile of the system. We are not considering software core organization this time, but for configuration profile idea to work we should program devices on all architecture levels except coordination a special but identical software code. This startup code should be capable of receiving setup commands from coordinator and reprogram device functions if needed. Once online all devices can be configured to current system application profile by the leader on coordination level.
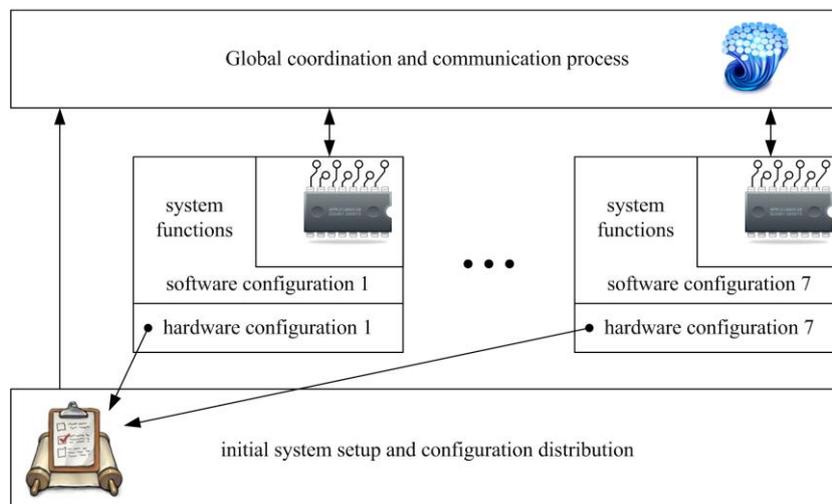

Fig. 2. General idea of configuring distributed control system.

First step to realization of introduced system architecture is presented on figure 3. It's an example of control board developed by beArobot team for Eurobot 2010

A. Vlasov and A. Yudin "Distributed Control System in Mobile Robot Application: General Approach, Realization and Usage" 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. 180-192.

competitions. We won't spend much time on describing all properties of the board as it would be enough information for a separate article. But what is to be highlighted is the need for a special interface board that is one of the key points of introduced architecture. It allows changing to be done only in the interface, using same control board, for all structural levels of the system and it gives capabilities to teach printed board design and circuit engineering without significant revision of the whole system mainly on the basis of mechanics tasks as they change each year of competitions.

Boards are based on AVR-core microcontroller and offer a wide range of communication capabilities which can be easily accessed with a number of connectors. No task specific parts are included onboard. Most of microcontroller ports' pins are traced to two special connectors meant to be standard for all other control boards with different core microcontroller. This standard allows a number of functional boards to be developed by team members while working on another project or use functional boards developed for other projects as well. Special effort was put in hardware configuration abilities of the boards with jumpers and soldered components, so that most of pins naturally used for onboard needs can be also accessed within standard connectors.
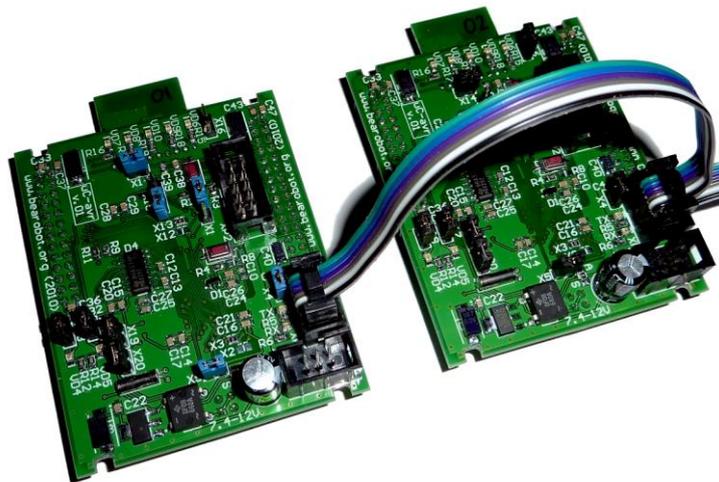

Fig. 3. Hardware control units – interconnected embedded boards example.

Optionally and mainly for debug purposes each board has a wireless connection capability. This is done to help developers have easy software access to each board without plugging in any cables. That could be useful whenever the whole system is installed on the robot but some debug processes are still carried out on individual devices. In normal operation wireless connectivity can be switched off by hardware or software means.

A. Vlasov and A. Yudin "Distributed Control System in Mobile Robot Application: General Approach, Realization and Usage" 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. 180-192.

4 Conclusions

This article presents the first step towards universal distributed control system for mobile robot applications. In the first place author expects this solution to be useful for student teams studying through Eurobot competitions. One of the far going goals and usages for the architecture scarcely described is forming of practical training methodology for future engineers.

Work on the subject is far from completion. Still, provided argumentation allows further development and ideas posited propose future research as a natural application of the system. Moreover if you look back at the beginning you'll see that research was the cause of even starting this work. But unfortunately we are still several steps before that main research itself. For now some auxiliary work is necessary to advance to outskirts of control science.

As the article is written while preparing for another Eurobot competition author expects to use results of this work in a real application to test ideas, to get solutions and to find improvements. Natural result of such work would be software and hardware realization which could become a subject of another article.

References

1. Simon, H.: The sciences of the artificial. 3rd ed., Massachusetts Institute of Technology, Cambridge (1996)
2. Eurobot, international robotics contest, http://www.eurobot.org
3. Colnarič, M., Verber, D., Halang, W.A.: Distributed Embedded Control Systems. In: Advances in Industrial Control series pp.62-63. Springer-Verlag London Limited (2008)
4. McCulloch, W.S., Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity. In: Bulletin of Mathematical Biophysics, vol.5, pp.115-133. (1943)
5. Norman, D.A.: The Design of Everyday Things. Basic Books, New York (2002)
6. Neumann, J.: Theory of Self-Reproducing Automata. 2nd ed., Edited and completed by Burks, A.W. University of Illinois Press, Champaign (1966)
7. Penrose, R.: The Emperor's New Mind: Concerning Computers, Minds, and The Laws of Physics. Oxford University Press, New York (1999)