LOCOMOTION AND RECONFIGURATION OF A MODULAR ROBOTIC CHAIN

Geanina Mihaela BOBOCEA[1], Constantin NIŢU[2], Bogdan GRĂMESCU[3]

Articolul prezintă un sistem robotic modular, proiectat pentru a fi capabil de reconfigurare, putând să se deplaseze prin păşire sau târâre. Sunt prezentate structura sa mecanică şi sistemul de control. De asemenea, este propusă o soluţie modificată de modul cu două grade de libertate, pentru care este analizată influenţa geometriei robotului asupra locomoţiei şi strategiei de auto-reconfigurare, prin simularea într-un mediu virtual.

The paper deals with a modular robotic system, which is designed for reconfiguring, in order to achieve both walking and crawling locomotion. Its mechanical architecture and control system are presented. A modified robotic module, with 2 degrees of freedom, is proposed, for which the influence of the robot geometry upon the locomotion and self-reconfiguration strategy is pointed out, by simulations performed in a virtual environment.

Keywords: Modular robotics, Autonomous system, Distributed control, Self-reconfiguration, Self-replication.

1. Introduction

Self-reconfigurable robots are complex distributed systems, consisting of several modules, with one, two or three degrees of freedom (DOF), as simple robotic units. These ones can be assembled in such ways, that form different mechanical structures, able to meet the requirements of the specific tasks concerning locomotion and self-reconfiguring. A self-reconfigurable modular robot is meant to adapt both to unstructured working environment and to malfunction of some own modules, by changing its configuration.

The existing modular robots are still laboratory achievements, built as demonstrators, but they proved to have real skills for locomotion and inspection in environments with strong variable geometric configuration, such as narrow spaces. There are many modular robotic systems, but a general classification can

[1] Ph.D. student., Faculty of Mechanical Engineering and Mechatronics, University POLITEHNICA of Bucharest, Romania, e-mail: bobocea_geanina@yahoo.com
[2] Prof., Department of Mechatronics and Precision Mechanics, University POLITEHNICA of Bucharest, Romania, e-mail:constantin.nitu@upb.ro
[3] Assist. Prof., Department of Mechatronics and Precision Mechanics, University POLITEHNICA of Bucharest, Romania, e-mail: bogdan.gramescu@upb.ro

be made by taking into consideration their shape and modules connection architecture: chain type and lattice type modular robots [1]. The chain-like robots are biologically inspired by snakes, worms or insects. Their modules are series connected or in tree-like structures, basically consisting in actuated joints, connecting mechanisms, power supply and control electronics. Anyway, from the commercial point of view, modular technology still need to demonstrate it could be better than conventional robotics.

An example of chain-type modular robot is *PolyBot* [2]. This system (G3 generation), developed by Palo Alto Research Center, is based on a cubic module with one DOF (rotation) and a hermaphrodite connection mechanism, which allows modules coupling both with parallel and perpendicular axes of rotation. For accomplishing the modules attachment to new locations, because the movement with respect to a touching neighbor module is not a trivial operation, photo-diodes and light-emitting diodes are used to determine the position and orientation of modules. Locomotion gaits of PolyBot range from a rolling loop, to a snake-like chain with sinusoidal movement, to a multi-legged spider or centipede construction.

Another example is *M-TRAN* [3], developed at AIST - Japan. Many configurations are possible with this robot: caterpillar, four-legged walker, etc. M-TRAN module is a two DOF homogenous one, with two rotations around two parallel axes. Reconfiguration occurs by repeating several elementary rotations for coupling to and detaching from a neighboring module.

SuperBot is one of the leading systems in the field, built by Polymorphic Robotic Laboratory – University of Southern California [4]. The module has a similar kinematics to M-TRAN, but with an additional DOF: rotation around the longitudinal axis. By help of this mobility, the modules can change configurations to enable different modes of locomotion, such as slither, crawl, walk, run, roll, climb, etc.

Even there is a limited field of kinematic options for a robotic chain, the above mentioned prototypes have had spectacular achievements, particularly M-TRAN and Superbot.

2. Robot structure

ROMAR (RObot Modular with Ability of self-Reconfiguring) [5] is a chain-type modular robot, built at University POLITEHNICA of Bucharest – Department of Mechatronics and Precision Mechanics. The first generation module consists of 3 basic elements which are cubes. It has 2 DOF: one rotation perpendicular to the longitudinal axis and the second around it. The external cubes can rotate within the range ± 90° (fig.1). Both rotations have significant importance for locomotion and reconfiguring. This design allows building of a cubic mesh, by dividing the robot working space into cubes having the dimension of its basic elements. The movements for reconfiguring or displacement are obtained by filling of the a priori defined cubic locations of the mesh with robotic elements, according to the control algorithm. The concept was previously verified by simulation of the components movement within Blender, a free software resource.

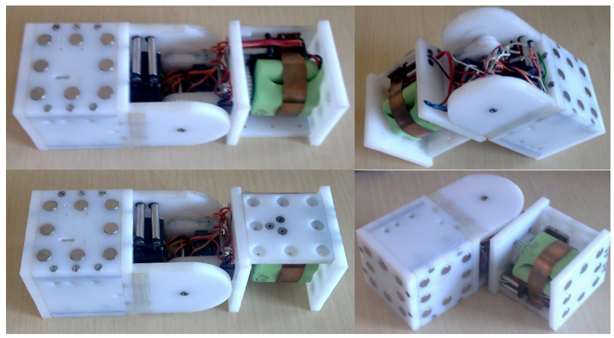
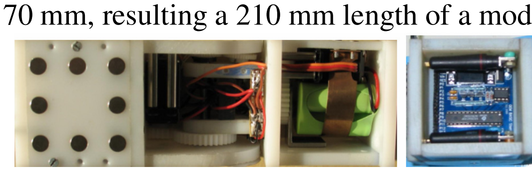


Fig.1. Module DOFs

The right side cube of the module (fig. 2, a), which performs the rotation around the longitudinal axis, mainly houses a RC servo type motor - MG995 (1 N.m) and a Ni-Mh battery. The rotation around the transversal axis is performed within central cube, which houses a RC servo of the same type. The left cube (fig. 2, b) is used to host the microcontroller (Basic Stamp 2 SX), RF communication modules, both from Parallax Inc. and the detaching mechanism. The connection between modules is accomplished by means of permanent magnets, while their detaching is helped by a leverage system, powered by a third motor. The size of an elementary cube is 70 mm, resulting a 210 mm length of a module.



a) b)

Fig.2. a) ROMAR module; b) Left-side cube

a) b) c)

Fig.3. Several configurations with ROMAR modules: four-legged walker (a), and self-replication into mini-chains (b), six-module caterpillar (c)

Several experiments with different configurations were successfully developed. Among these: four legged walking robot (fig. 3, a), crawling locomotion with a chain of six modules (fig. 3, b) and Self-replicating from open chain structure, consisting of 6 modules, to 2 autonomous structures consisting of 3 modules (mini-chains) and their crawling locomotion.

Starting from this first version of the ROMAR module, a newer one is proposed for this robotic system regarding module's shape and structure.

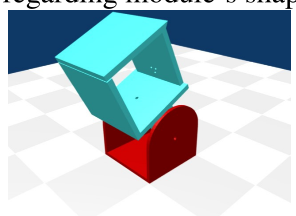


Fig.4. ROMAR II module in a 3D simulation environment

ROMAR II module (fig. 4) consists of only 2 cubic elements. Similar to the previous version, the number of DOFs is the same: a rotation around a transversal axis and another one around the longitudinal axis. Both have the same range, ± 90°. The new module keeps the main components of the first version: microcontroller Basic Stamp 2SX, RF wireless communication and RC servos. The new cubic elements have a 100 mm edge, so the resulting module size is 100 mm x 100 mm x 200 mm. These dimensions were adopted for the possible use of a more accurate, robust and stronger servomotor, which also requests a driver board and a larger capacity battery to be placed inside each module. Changing the type of connection between modules is also considered, by replacing the permanent magnets with a mechanical system.

3. Robot modeling

The main goal of developing the dynamic model of the robot is to check, before building it, if it is able to perform the motion according to a command sequence of different active joints. The most demanding situation is the wormlike crawling as in fig. 3,c. This type of locomotion can be decomposed in 2 basic steps. For example, at the step 1, the segments 12 and 23 are getting up with an angle θ, in order to achieve a triangle (fig.5). It was considered that all the segments have the same length l, and the other geometric parameters are denoted by: lG1= lG2= lG3 = lG – the distances from the axes 1, 2 and 3 to the gravity centers of the modules 01, 12 and 23.

The equivalent moment of inertia Jr, for the situation in fig. 5 is [5]:

$$J_r = 2J_G + ml_G^2 + 2m(l^2 + l_G^2) - 4mll_G \cos^2\theta + 4ml^2 \sin^2\theta \quad (1)$$

where: J_G – the moment of inertia of the modules 12 and 23; m – module mass.

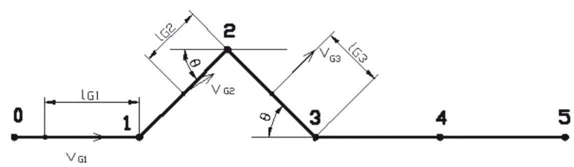


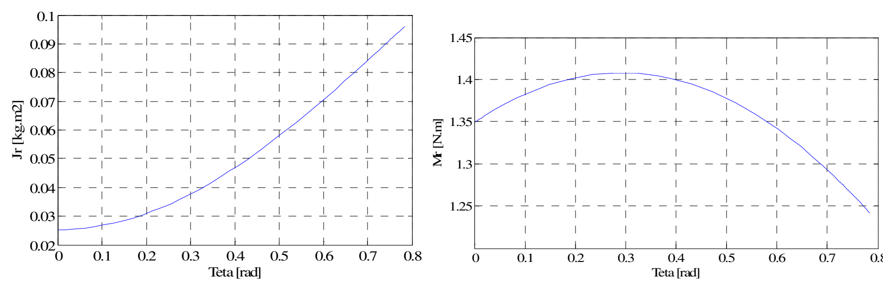
Fig.5. The kinematic chain for the step 1

The equivalent torque at the actuated joint, given by gravity and friction forces, was calculated from virtual work of the resistant forces:

$$M_r = 2\mu mgl \sin\theta + mgl_G \cos\theta + mg(l - l_G)\cos\theta + M_{f0} \tag{2}$$

where: μ – coefficient for the friction between module 01 and the displacement surface; g – the gravity acceleration; M_{f0} – friction torque in the motor bearings.

Variations of the inertia and of the equivalent resistant torque (fig.6) were calculated in Matlab, by help of (1) and (2). The interest range is up to 45º (less than 1 rad). A similar evaluation for the step 2 (fig. 7) leads to smaller values of Jr(θ) and Mr(θ).



a) b)

Fig.6. a) Jr(θ) diagram for step1, b) Mr(θ) diagram for step 1

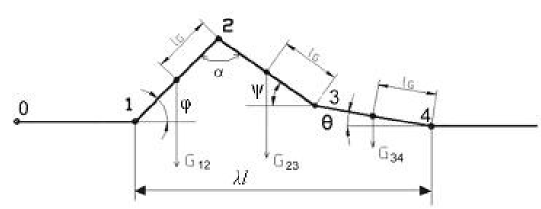


Fig.7. The kinematic chain for the step 2

So, a simple model of the actuating joint (fig.8) was developed in *20-sim*, in order to check the RC servo capability to move the robotic chain in the worst case of its position. In this model, the RC servo construction and operation were taken into consideration. The pulses with period of 20 ms are interpreted as a reference value (step generator), compared to the actual position information, provided by the position sensor (potentiometer), as feedback. The dc motor is driven by PWM signals, so a current control model was adopted. For the mechanical part, the rotor inertia, gear ratio and the robot segments inertia, as well as a constant resistant torque at the maximum value were also introduced.

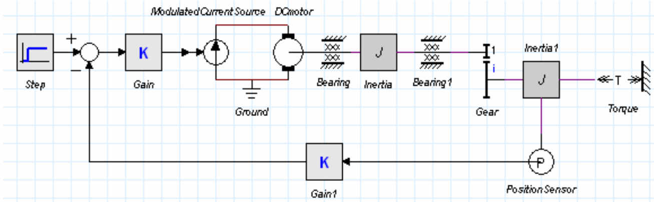


Fig.8. Robot simplified model in 20-sim

The model response (fig. 9) shows the capability of an active joint to move at least an additional module, in an acceptable period of time.

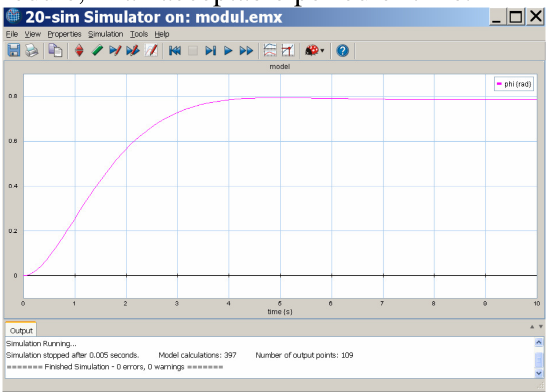


Fig.9. System response of the model in fig.8

4. Locomotion and reconfiguring strategy

The locomotion abilities of the robot ROMAR I are the ones associated with the shapes in fig.3: a) four-legged walking structure; b) three modules crawling structure; c) six modules chain structure with snake-like (or worm-like) movement. A self-reconfiguring procedure for transforming the four-legged structure into an open chain was conceived and tested, as well as the locomotion of the above mentioned structures. The main difficulty of the ROMAR I locomotion is related to walking, because the legs don't have the adequate mobility and must pull along the floor.

ROMAR II is an attempt to correct this deficiency. Module design allows the same ability of building a cubic mesh, by dividing the robot working space into cubes, like the previous version. The concept was verified by simulation within *Blender 3D*, a free software resource. Several movement and reconfiguration possibilities have been investigated, with a different number of modules, in order to determine the mobility of different structures with ROMAR II modules.

Mobility of six modules

Compared to the four-legged walker configuration with six modules of the first version, the similar ROMAR II six modules configuration can achieve the locomotion, but the self-transforming from a four-legged walker to an open chain structure is not possible.

Mobility of seven modules

By adding an extra module to the four-legged walker configuration with six modules, the possibility of reconfiguration is achieved. Important steps in the reconfiguration process from a four-legged walker seven modules configuration to a snake-like configuration are shown in fig. 10.

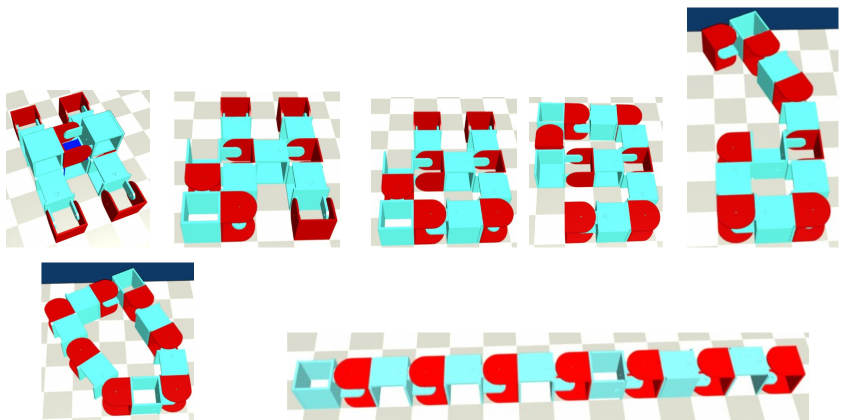


Fig.10 Reconfiguration steps from a four-legged walker to snake-like structure with seven ROMAR II modules

The last image in fig. 10 shows seven modules in open chain configuration with not properly oriented joints to achieve vertical undulations like a caterpillar. Thus, the last reconfiguring steps are to bring one by one the joints in the right positions. This task is achieved by rotating pairs of neighboring modules around their longitudinal axis.

The reconfiguration process of the four-legged walker to open chain is simpler for ROMAR II, than in the case of ROMAR I. Only one reconfiguration step is outside the cubic mesh of the working, whereas three similar steps outside the cubic mesh were needed for the reconfiguration of ROMAR I modules.

Mobility of eleven modules

An eleven modules version of the four-legged walker has also been designed and simulated in *Blender*. In this one, each leg is built two modules, the structure gaining a better mobility with the extra joints.

Reconfiguration, as it can be seen in fig. 11, is achievable by help of the extra module on top, which does not contribute to locomotion. The number of steps necessary to accomplish the legged robot transforming into an open chain is larger than for the structure with seven modules, but the last steps of reconfiguration are identical. As it can be observed, this reconfiguration requires more steps outside the cubic mesh compared to the one of the four-legged walker with seven modules, which needs only one, but this small disadvantage is compensated by the increased mobility of this structure.

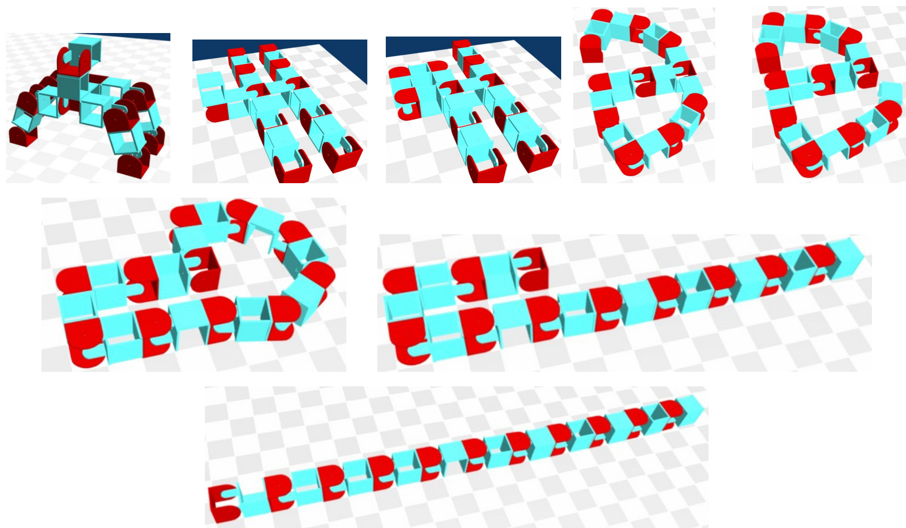


Fig.11 Reconfiguration steps from a four-legged walker to snake-like structure with eleven ROMAR II modules

More outcomes than the one presented above can be achieved by reconfiguration of this structure. If the reconfiguration process is stopped in the mid-step, the structure can be divided in open chains: two snake-like structures with four modules, and a snake-like one with three modules. Each structure is able to move independently by vertical undulations.

5. Robot control

Basically, the same control system is valid for both versions of the robot – ROMAR I and the new ROMAR II [6]. The only difference lies in adapting the control algorithm to the different geometrical requirements, concerning their structures and shape.

Robot control was hierarchically developed in 2 levels (fig.12). The lower level is the local control system, implemented on a Basic Stamp microcontroller for each module, responsible with basic motions and connection/ detaching operations.

This microcontroller was chosen for the easiness and quickness of the prototype development. The upper level refers to the global control system, which provides the information concerning the required configuration of the modules, for achieving a certain displacement type or reconfiguring. This one was implemented on a host computer, enhanced with a digital I/O board, for use with LabVIEW programming environment. The two control systems communicate by means of wireless devices (Parallax 433 MHz RF Transmitter/Receiver).

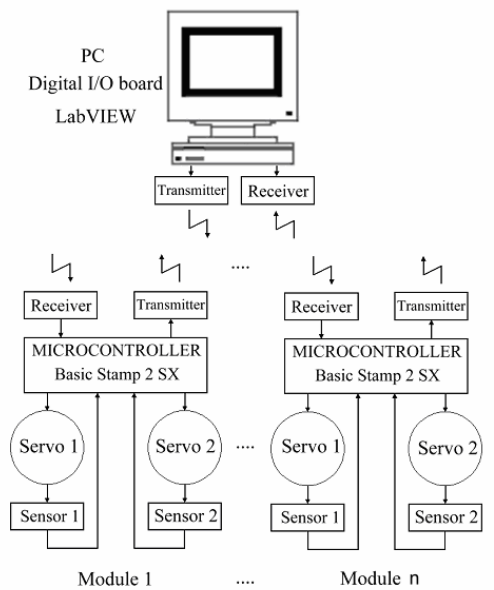
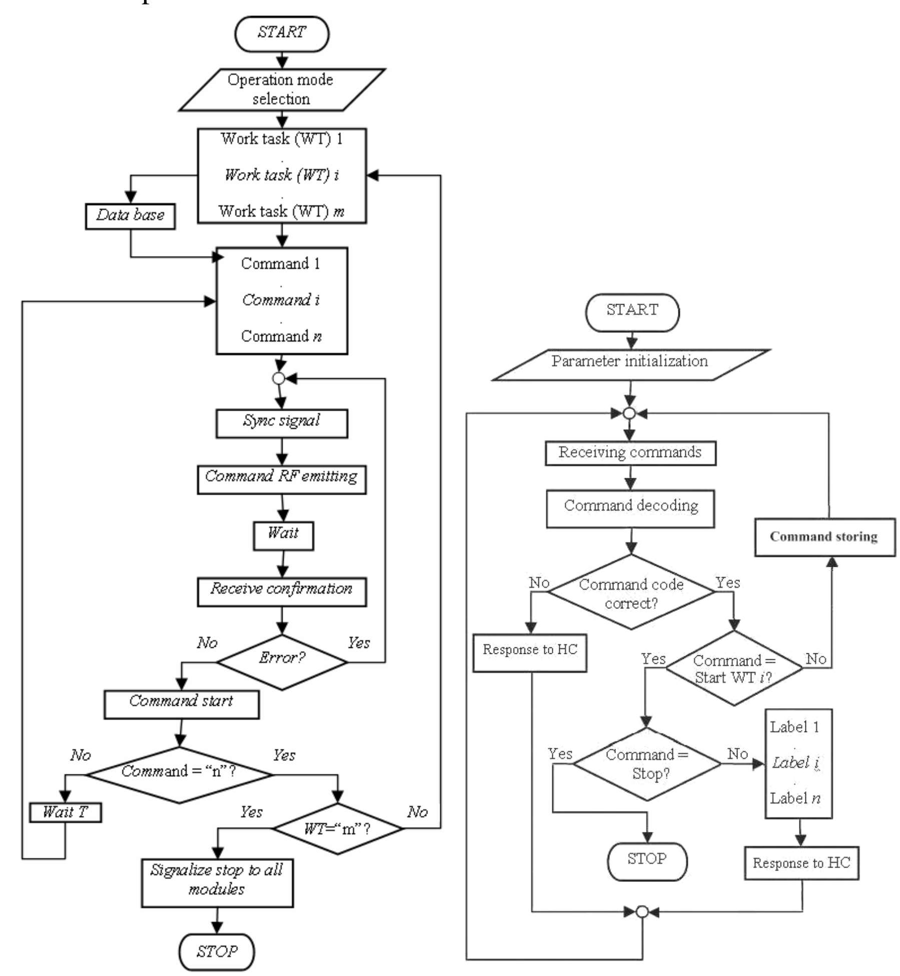


Fig.12. Control of the robotic system

The flow chart of the host computer program is shown in fig. 13.a. The first step of the program is the parameter initializing: variables types and constants, number of robot modules, the spatial position of each module, the period of time for achieving the needed configuration, initially determined by using geometrical and dynamic equations.

The second step consists of setting the needed work task, by choosing it from a list of possible ones. Then, the program calls a data base (DB) and extracts the commands list for the work task. This list must be transmitted to the modules, starting with the first command.

The synchronization of the command receiving by each module is necessary. For this purpose, the host computer is sending a RF synchronization signal before the command, announcing the modules about the transmission start. Then, the first command is emitted, the host computer waits a period of time for receiving the confirmation from the robot modules. In the case of a transmission error, the command information will be transmitted again until there will be no errors. At that moment, the program verifies if the command information is the last one from the list. If there are more to be sent, the program waits a period of time and return to the next command in the list. If there are no more commands to be sent and no more work tasks, the host computer program emits a stop signal to all modules and stops itself.



a) b)

Fig.13. Flow charts: a) of the host computer program
b) of the robotic module program

The flow chart of the program implemented on the robotic module is shown in fig. 13.b. From the structure point of view, the programs on each the robot modules are identical. The significant differences are regarding parameter initializing with the specific calibrated values for the motors, and the specific program execution of the received commands.

After the program is started and the parameters are initialized, the module program receives the synchronization signal and the command information.

If the command is not addressed to a specific module, the program emits a response signal for the host computer, waits a determined period of time and goes to the receiving commands subroutine again.

If the command is addressed to the module, the program decodes it and verifies if the command code is correct. If the answer is no, the module emits an error signal to the host computer, waits a determined period of time and go to the receiving commands subroutine. If the command code is correct the program verifies if this is a general stop command. In the affirmative case the program will stop, or in the case of a specific command the program will continue.

If the command signal is correct, but other module sends error signal, the host computer will be forced to emit the command again. The module must avoid multiple executions of the same command. That's why the program compares the new command with the last one, and if this is the same, it memorizes the command and sends an answer to the host computer waits a determined period of time and goes to the receiving commands subroutine. Only if the command is a new one, the program will go further.

The module program has a list of execution subroutines, identified by the command codes. The selected one is executed, the module sends an answer to the host computer, wait and go to the receiving subroutine again.

The communication between the host computer and the robotic modules is based on a pair of wireless devices – RF emitter/receiver, mounted on each module and one pair on the host computer. For this project, the chosen communication solution is LPD433 (Low Power Device 433 MHz). This devices are frequency modulated (FM) with a maximum output of 10 mW (legal domain). Emitter/receiver modules are using this standard and are fully compatible with BS2SX.

6. Conclusions

The development of a modular self-reconfigurable robot is possible with constant technology advancements, which enable achievements in manufacturing miniaturized parts, sophisticated control and power supplies for autonomy enhancing. In these circumstances, mechanical architecture approach, with its strength and weakness is the key point of a successfully development of such a robot. This means the modules mobility and robot geometry are influential factors in locomotion and reconfiguration strategies.

R E F E R E N C E S

[1] *M. Yim, Y. Zhang, D. Duff*, Modular robotics, IEEE Spectrum, February 2002, pp.30-34

[2] *M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans*, Modular reconfigurable robots in space applications, Autonomous Robots, Vol. 14, No. 2-3, 2003, pp. 225-237, Springer Netherlands.
[3] *H. Kurokawa, E. Yoshida, K. Tomita, A. Kamimura, S. Murata, S. Kokaji,* Self-reconfigurable M-TRAN structures and walker generation. Elsevier Robotics and Autonomous Systems 54 2006, pp. 142–149
[4] *W.M. Shen*, Self-Reconfigurable Robots For Adaptive And Multifunctional Tasks. In Proc. 26th Army Science Conference, Florida, USA, December 2008
[5] *C. Niţu, B. Grămescu, S. Niţu, C.D. Comeagă*, Self-reconfiguring Modular Robot, In Proc. of 2008 IEEE Int. Conf. on Automation, Quality and Testing, Robotics, Tome II, pp. 320-324, 2008, Cluj Napoca, Romania
[6] *B. Grămescu, G.M. Bobocea, A.O. Trufaşu, C. Niţu,* Motion Generation for a Modular Robot. In Proc. of 2010 IEEE Int. Conf. on Automation, Quality and Testing, Robotics, pp. 292 – 296, 2010, Cluj Napoca, Romania.