Optimization of fast moving robots and implementation of I^2C protocol to control electronic devices

Fernando Ribeiro, Pedro Silva, Ivo Moutinho, Vítor Silva, Nino Pereira

Grupo de Automação e Robótica, Departamento de Electrónica Industrial,
Universidade do Minho, Campus de Azurém, 4800 Guimarães, Portugal
fernando.ribeiro@dei.uminho.pt, ivo@aeiou.pt,
pedro_silva@portugalmail.com, vic_silva@mail.pt, ninopereira@clix.pt

Abstract. This paper briefly describes the main changes in the robots from Minho Team since the last RoboCup. Work has been carried out on the robots in order to constantly improve their capabilities, based on the experiences from previous participations. The main topics to deal in this paper are related with the I^2C new communications protocol recently implemented in the robots, a description of the whole electronics system based on a new diagram which contains all the devices used in the robots, the inclusion of new faster motors, a description of the robot game play strategy, some operating system remarks which are related to some requirements of our actual system, and the main conclusions related to the work carried out so far. The use of the I^2C protocol forced the team to change all the electronic boards and these were completely redesigned in-house, tested and implemented on the robots.

1. Introduction

Minho Team develops a robotic football team since 1998 and has been participating on RoboCup since 1999 [1][2][3][4]. On this team work one professor, one mechanics technical staff and four under-graduate electronics students. This team develops the robots from scratch, designs all the electronics, the high- and lo-level software and the mechanical structure. The main areas of research or interest from the students are related with electronics design (since the team belongs to an Electronics Department), computer vision software and control algorithms. Even though last RoboCup participation was successful, for this year challenge, the team decided to improve many aspects of the robots starting with new faster and more reliable motors, the implementation of a new electronics communication protocol I^2C, and subsequent redesign of all the electronics boards, not just to implement the I^2C protocol but also for optimization of the boards. Other small changes have been implemented but these are not as relevant as the previous ones, and will be described in future work.

2. Mechanical Aspects

The mechanical structure of each robot did not change dramatically since last year but only small fixtures. All the robots are similar (even the goal keeper), with a cylindrical shape on the bottom half and cylindrical shape on the top half.

Underneath the robot, three sets of motor plus planetary gear head plus encoder are coupled directly to omni-directional wheels from Kornylak are assembled. The level above contains the kicker device, four lead acid 12V 7Ah batteries and also the motor control and power electronic boards. On the next level are located the kicker electronics board, plus the ball handler device. The next physical layer holds the robot computer. The top layer has the camera and omni directional mirror. The first levels are described in **Fig. 1**.

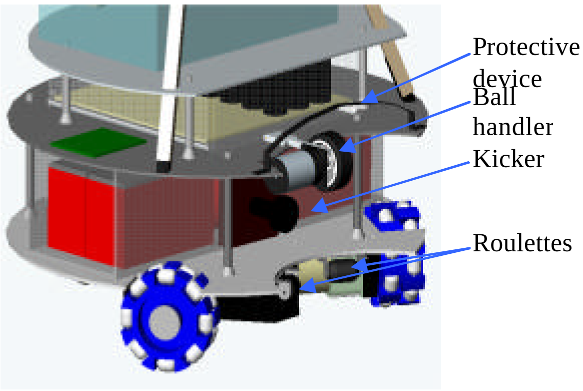

Fig. 1. CAD drawing of the robot’s bottom half (three levels)

This type of aluminium construction structure proved to be robust. The heavier parts were all placed at the bottom part of the robotic platform so that the centre of gravity was as low as possible. The vision head is placed on the highest position to increase the field of view. It consists of a colour camera facing upwards onto a parabolic mirror facing downwards. The tripod which supports the vision head is also made in aluminium so that it gets very light, and it is also easy to dismount should it be necessary. This structure was previously designed in a three dimensional CAD to allow part position planning, so that any part replacement, when time is critical, could be carried out very quickly and in most cases without the need to remove any screw. It can be seen in **Fig. 2** the whole robot designed in the CAD system and by its side a picture of the final robot.

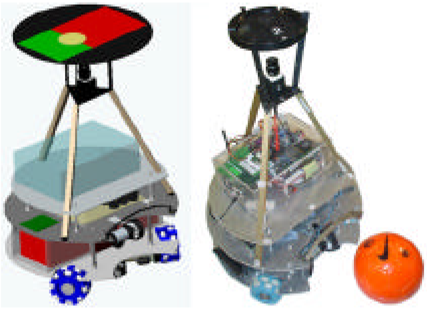

Fig. 2. CAD drawing of the complete robot and final real robot

For locomotion these robots use three omni wheels coupled to D.C. geared motors of 150 Watts from Maxon, with a 500 pulses per turn encoder. A ball handler is used to make the ball spin, geared by a DC motor coupled to a rubber wheel. In order to kick the ball a magnetic kicker is developed completely developed by the team. It proved to be very strong and it consists of a magnetic coil with a movable iron core which is pushed towards the coil centre when electric current passes through the coil.

Two infrared sensor devices are used in order to detect when the ball is in conditions to be kicked, so that the impact force is optimized. The four lead acid 12V 7Ah batteries are easy to replace and are placed two on each side. One battery is used to power the computer and the others power all the electronics, motors, kicker, etc. The omni-directional wheels are easy to dismount should they need replacement.

3. Hardware System

Until RoboCup'2004, all the devices on the Minho Team robots were controlled through the computer parallel port, where 16 devices at maximum could be accessed (**Fig. 3** describes the old configuration). This solution proved to have enough speed to control the devices but showed low immunity to Electro Magnetic interferences, due to the use of flat cables. Moreover, this topology is very much limited what concerns the maximum cable length and number of addressable devices.

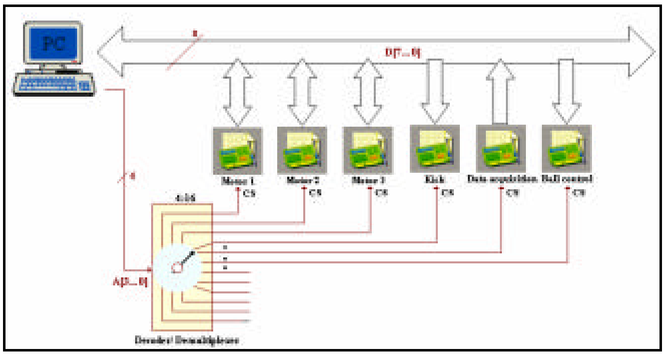

Fig. 3. Computer Parallel port device control

It was urgent to change to another solution to overcome these problems at low cost. A study was carried out to choose a protocol and the main criteria were:

- Immunity to electromagnetic interferences (EMI)
- High speed communications
- Protocol efficiency and communication error control
- Possibility of addressing a larger number of devices
- Increase communication distances
- Implementation simplicity
- Low cost

The most important protocols were compared and a table was produced with a comparison with the main advantages and disadvantages.

Table 1. Advantages and disadvantages of each device communications protocol

UART	CAN	USB	SPI	I2C
• Well Known • Cost Effective • Simple	• Secure • Fast	• Fast • Plug & Play HW • Simple • Low cost	• Fast • Universally accepted • Low cost • Large portfolio	• Simple • Well known • Universally accepted • Plug & Play • Large portfolio • Cost effective
• Limited functionality • Point to point	• Complex • Automotive oriented • Limited portfolio • Expensive firmware	• Powerful master required • No Plug & Play SW – Specific drivers required	• No Plug & Play HW • No “fixed” standard	• Limited speed

After analysing this table it was decided that the protocol that best suits this team needs was the I^{2}C.

3.1 I2C Bus protocol

This type of BUS physically consists of 2 active wires and a ground connection. The active wires are both bidirectional and correspond to the Serial DAta line (SDA) and the Serial CLock line (SCL).

Every component hooked up on this bus has its own unique address no matter it is a motor controller board, an infra-red sensor, a CPU, LCD driver, memory, or any other complex function chip. Each chip can act as a receiver and/or transmitter depending on its functionality. While an LCD may act only as a receiver, a motor controller board can be both transmitter and receiver depending whether the user needs to input the desired velocity or read the electric current on it.

The I2C protocol specifies that the Integrated Circuit that initiates a data transfer on the bus is considered the BUS MASTER. From then on, all other devices are regarded to as BUS slaves. Even though this protocol allows more than one BUS MASTER's, generally only microcomputers can do that, in this robot system happens precisely that; only one is the Master (the Computer motherboard) and all other devices are SLAVES.

This is attractive for portable equipment and systems that run on batteries due to its low current consumption, high noise immunity and wide supply voltage range.

The main advantages of the I2C protocol are:

- Only two bus lines are required
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times

- It is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer
- Serial, 8-bit oriented, bi-directional data transfers can be made at up to 100 kbit/s in Standard-mode, up to 400 kbit/s in Fast-mode, or up to 3.4 Mbit/s in High-speed mode
- The number of ICs that can be connected to the same bus is limited only by a maximum bus capacitance of 400 pF
- ICs can be added to or removed from a system without affecting any other circuits on the bus
- Fault diagnosis and debugging are simple; malfunctions can be immediately traced
- Software development time can be reduced by assembling a library of reusable software modules.

The actual solution implemented is described in **Fig. 4**.

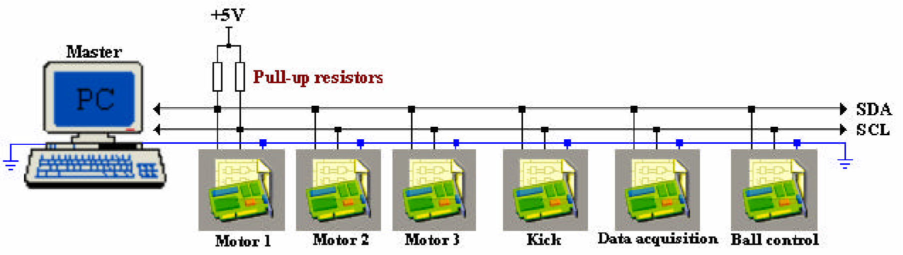

Fig. 4. Diagram representing the I^{2}C protocol implemented on the robots

This solution reduces dramatically the cabling on the robot simplifying its assembly and its debugging practices and above all reduces electric noise. More information on I2C-bus protocol can be read in [5]

3.2 Hardware Diagram

The devices function was already described in [3] and [4], but the electronic boards had to be completely redesigned, built and tested due to the use of the I2C protocol. Furthermore, some optimization and improvements were carried out on these boards. The whole Hardware system works around a low consumption computer motherboard VIA M10000 running at 1GHz speed, with 256 Mb memory, and a Flash of 512 Mb. A PCI bus frame grabber bt848 based is plugged in a slot, where to an analog colour camera sends the image. A USB wireless network adapter 802.11a based is used for radio frequency wireless communications between the robots. A 7 inches VGA touch screen is now part of the robot. A DC-DC converter powers the computer motherboard being fed by a 12V 7Ah lead acid standard off-the-shelf battery.
As previously described, an I2C protocol takes care of the communication between all the devices and the computer motherboard. A Ball handler control board takes care of

the motor which spins the ball, and a data acquisition board reads many states of the robot like each motor current, encoders, battery voltage, etc.

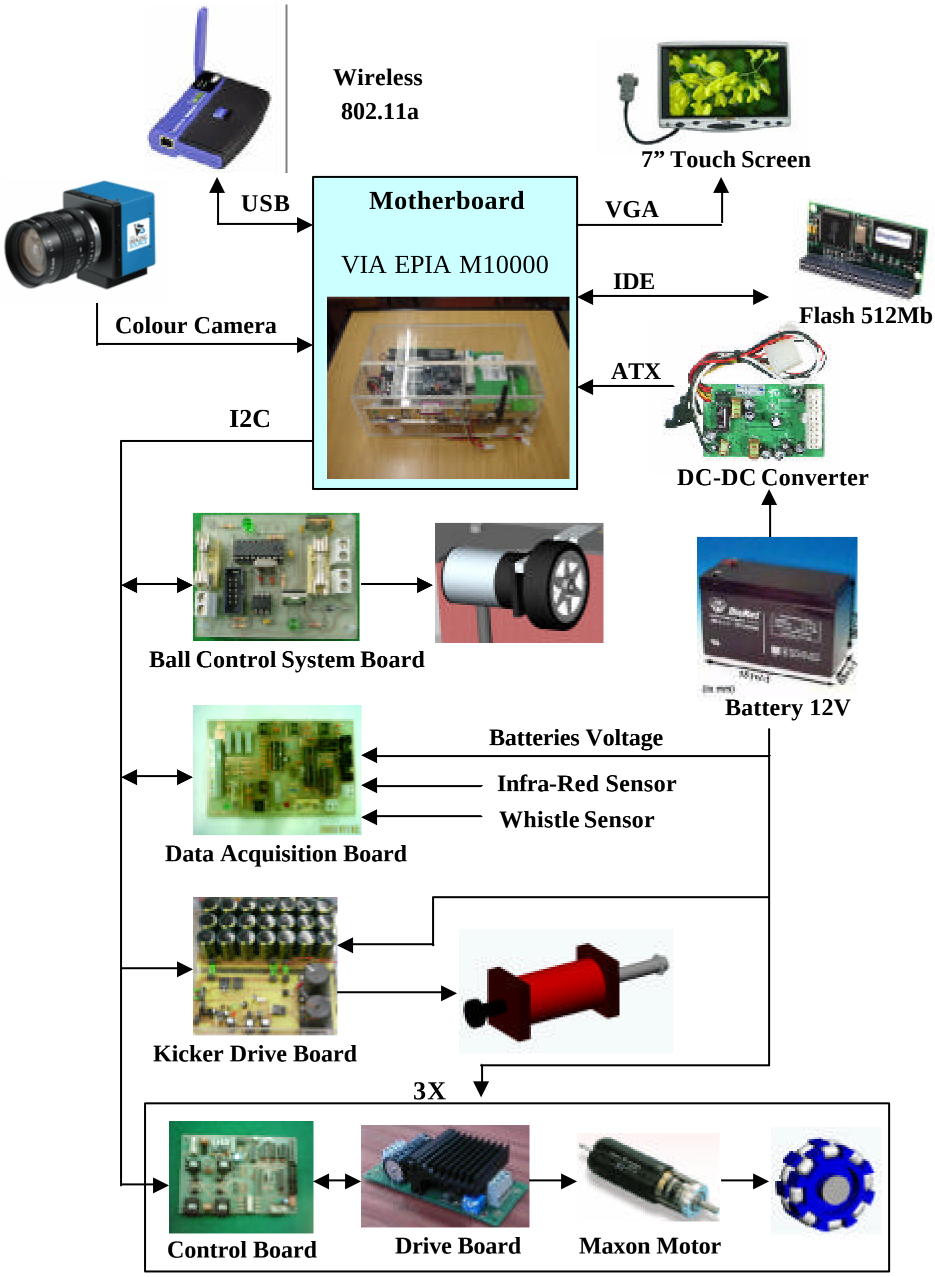

Fig. 5. Robot’s Hardware Diagram

A motor control board receives commands through I2C protocol and sends it to a motor drive board. All these electronic boards were developed by the team in the laboratory. The new motors are now 150W Maxon DC RE40, with a Maxon GP42C planetary gear head and a digital encoder HEDS55 with 500 pulses per turn and TTL compatible. These motors are very reliable, fast and its consumption is lower than the previous motors used, allowing the batteries to last longer.

4. Software Architecture

No matter how good the hardware is, it is the software that will make the robot play football. The choice of the best operating system, computer programming language, and software optimizations are factors to take into account very seriously in order to have successful player robots. The faster the code runs, the more efficient it can play.

4.1 Operating System and programming environment

Linux based operating system proved to be one of the most reliable operating systems for the RoboCup challenge robot computers. The operating system used by this team is Mandrake 10.1, kernel 2.6.4. Some modules needed to be installed, to run our software: *i2c-dev* (to use I2C devices), *i2c-viapro* (necessary for our motherboard), *bttv* (for the frame grabber) and *acx100_pci* (for the wireless network board).
Due to the reduced amount of disk space, XWindows is not installed and even the minimum Linux installation is used so that only the necessary processes run on the background.
The programming language for the whole application is C and the compiler used is *gcc* (version). The following libraries are used to compile the code: *svgalib 1.4.3* (for the graphics mode), *libmsock* (for TCP/IP client-server applications), *pthread* (to allow the use of threads), the *msock* (for network purposes) and the m (for some mathematical calculations). All software has been developed by the team and consists of routines files, program files and configuration text files. Each software area has a name associated:

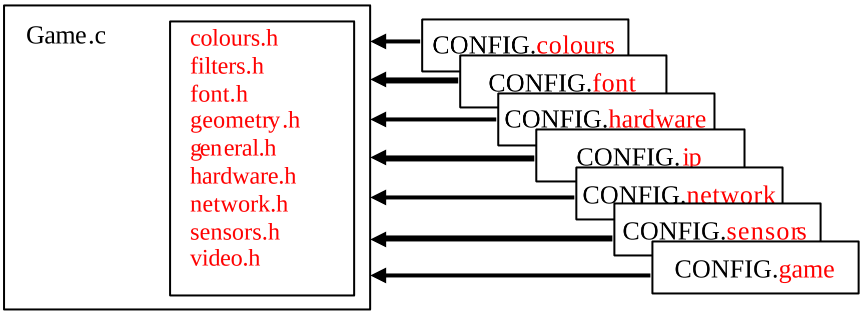

Fig. 6. File structure of the GAME program that the robots run

Each of these files has a text format configuration file, which contains variables values by default for the program. If a variable needs a new setup value, this text file is easily editable and there is no need to recompile the application.
The final program “GAME” deals with all routines files, and configuration text files.

4.2 Computer vision description

The vision system has been continuously upgraded, since this is one of the most critical parts of the whole system.

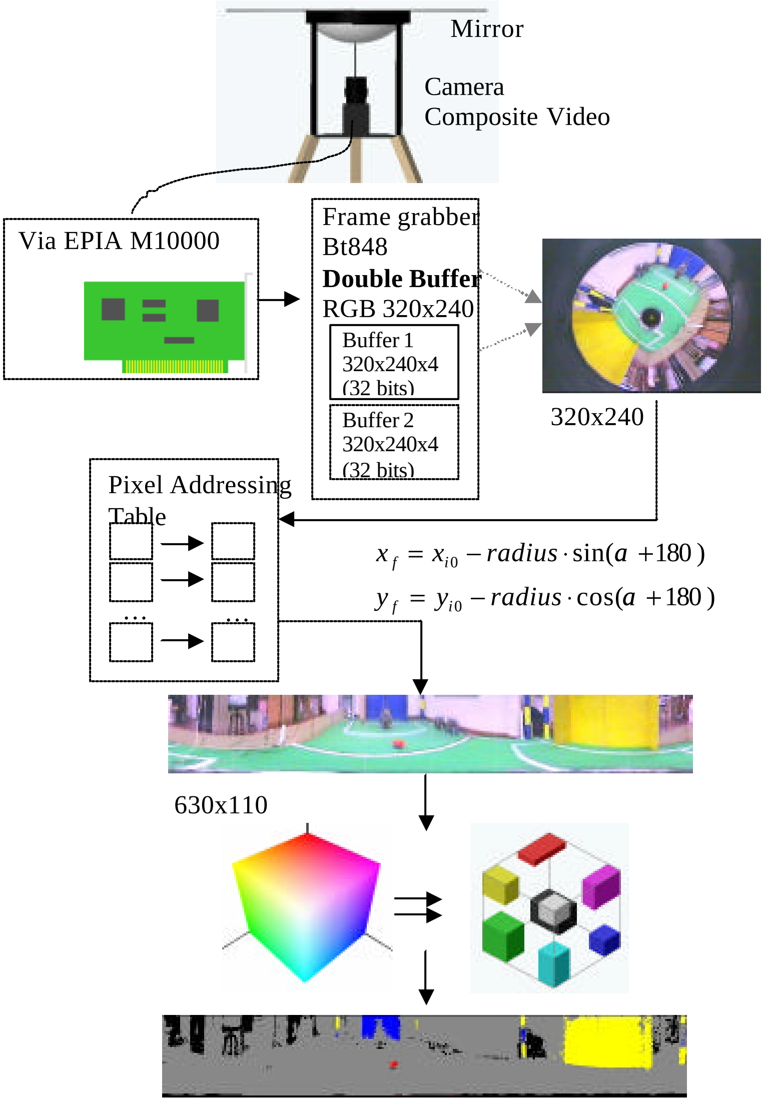

Fig. 7. Robot’s Vision System

Each robot has its own vision system, which consists of one composite video camera pointing upwards a mixed shape hyperbolic/conic mirror [4][6]. This mirror reduces the image but increases the real area captured. The conic part of the mirror avoids having the farther objects to look so small. The camera is linked to a PCI bus bt848 based frame grabber plugged in a Computer slot. Double buffering (odd and even frames) is used and the image size is 320 x 240 pixels. The captured format is RGB 32 bits to keep parity with the computer memory, so that the image takes no time to be displayed on the screen (it is a direct memory copy). The image grabbed is not in a desired geometric shape for computer processing, mainly because part of the image is upside down and trigonometric calculations are needed to percept real information. These trigonometric calculations are very processing time consuming. Therefore this round shape image grabbed is transformed into a rectangular image, through a pixel addressing table which is previously created before the program starts. This table contains the physical pixel address of each pixel for the rectangular image. The table values are calculated only once, but the transformation for all pixels is carried out for every frame. Even though, this Polar to Cartesian coordinates transformation takes less time than the grabbing time for one frame.
After the rectangular shape video image being displayed, a colour filter is used in order to consider only the relevant pixels. This reduces the processing time and simplifies the entities search on the image.
The filter consists of replacing a certain RGB sequence by one colour stored in a three dimensional table, whose indices represent the R, G and B. The cube on the left contains all the possible colours and the one on the right contains the colours which will replace the pixels on the image. The grey colour [127,127,127] has a special meaning as “no relevant colour” for the processing. This cube/table has 256 x 256 x 256 values and this indexing mode is the fastest way to access the expected colour.
In order to optimize the entities search, the vision routines have to be extremely optimized. One way to reduce the area inspected is by removing from that search the areas where the ball or the goals will most likely never be. For example, the ball will never appear in the top quarter of the image, and the goal will never appear in the bottom quarter of the image. It only makes sense to look for obstacles in the second quarter from the bottom of the image. Obstacles in the top half are very far from the robot and therefore are not processed in those regions.

5. Game Strategy

Even though it is the same software that runs on all robots, each robot has its unique behaviour on the field. This team plays with 4 robots: one Goal keeper, one defender, one forward and one striker. Sometimes robots have to be taken out of the field either for faults/cards or for break down. This team uses the number of players on the field to influence their behaviour on the field, like in a real human football game. For example, if only one robot player is on the field, it becomes a goalkeeper. With two players on the field, one will be Goal Keeper and the other forward. With three players, the third will become Defender and with four, the fourth will become a Striker. The order of behaviour is then: 1) Goal Keeper, 2) Forward, 3) Defender and

4) Striker. Each robot's behaviour has its unique sequence and that is described in the chart below.

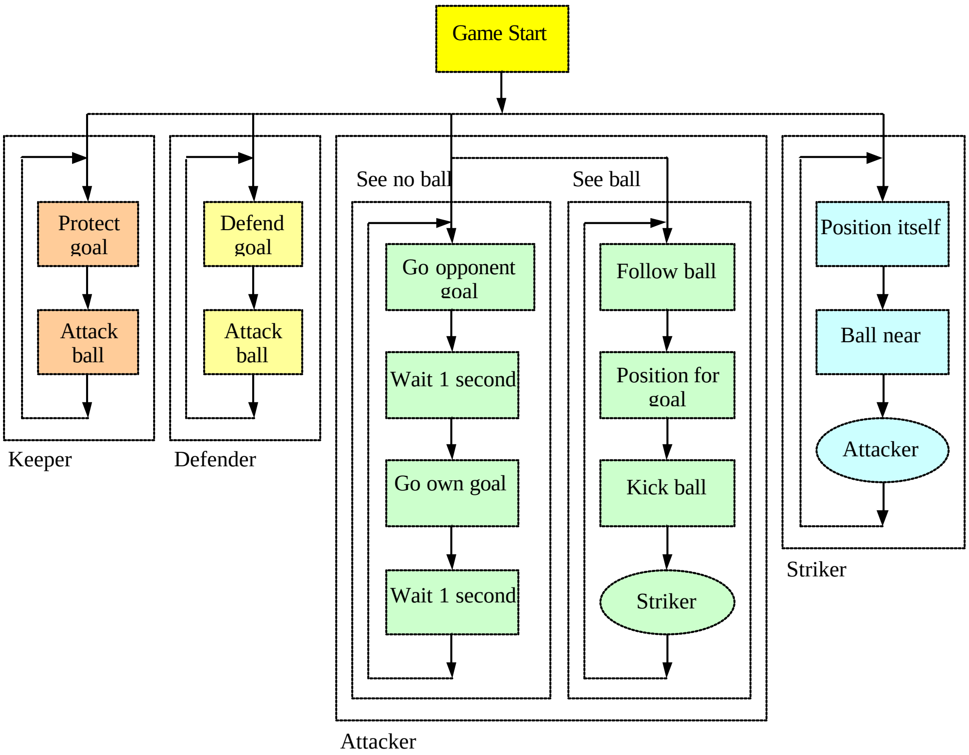

Fig. 8. Tasks sequence for robot behaviours

The *Striker* does not behave as a forward otherwise two players would go towards the ball. It consists of a player which positions itself in a clear line between the opponent goal and the Forward player. If the Forward has no clear line to kick to the goal, it will pass the ball to this Striker which will do that. This new type of player will not run after the ball but it will position itself in an easy location to score.

6. Conclusions

It is important to point out that this work carried out for the last few years is not just a research project, but an engineering and educational project. The main objectives are: a) to pursuit research in electronics, computer vision and control, b) to teach the students a practical case of a complex mechatronics system (where they have to plan solutions and sort out problems related with mechanics, electronics and software engineering), and also to make the students build such complex robots to prove in practice what they study in theory. If the result is successful it means they will be good engineers. The topic of football serves as an argument to involve students more easily in their work, despite the solutions having to be achieved successfully.

When many changes are implemented in the robots in the same year, there is the risk of failing to test them all, and it is more difficult to debug the system after all. Therefore, only a small number of changes should be performed at a time and preferably not related with each other. This year, this team decided to change the devices communications protocol to I2C and the DC motors to faster ones. Changing from a star to bus topology, reduces cabling and electric noise. This protocol change forced to redisign and to build all the electronic boards to comply with the I2C. The computer motherboard in use already had I2C embedded simplifying slightly the electronics in each control board. The software development was not difficult although the learning curve was slow at the beginning.

Now, all the devices are sequentially linked in a Bus topology with unique addresses, allowing the easy inclusion of more devices, without having to reconfigure more hardware and with simple source code to talk to them.

As stated in [4], in RoboCup football games, the time a robot takes to reach the ball is of extreme importance. The faster it gets the ball the more chances it has to score. Regardless of the 3 wheel drive configuration already used, this team needed to move faster towards the ball and goals. With these new Maxon mo tors, the robots can move up to 3 times faster, being necessary to change the control algorithm. Moreover, the new motors core will run at lower speeds making the energy consumption lower.

Acknowledgements

The team would like to thank the Algoritmi Research Centre, Department of Industrial Electronics, School of Engineering and University of Minho for all support. This Project is partially sponsored by the Portuguese Science and Technology Foundation of Ministry of Ciência e Ensino Superior under the research project “Development of Robotic Football Team for participation in the RoboCup (Middle Size League)”, POSI/ROBO/43892/2002.

References

[1] Sérgio Sampaio, Carlos Machado, Ilídio Costa e António Ribeiro, "Patriarcas/Minho football team", RoboCup_99 Robot World Cup Soccer Games and Conference IJCAI’99, Teams Description Small and Middle Leagues, páginas 175-180, http://www.ep.liu.se/ea/cis/1999/006/28, ISSN: 1401-9841, Julho 1999, Suécia.

[2] Fernando Ribeiro, Carlos Machado, Sérgio Sampaio, Bruno Martins, "MINHO robot football team for 2001", RoboCup'2001, Seattle Conference & Trade Centre, Team Description Paper, Augost 2001, Seattle, EUA.

[3] Fernando Ribeiro, Paulo Braga, Jorge Monteiro, Ivo Moutinho, Pedro Silva, Vítor Silva, “New improvements of MINHO Team for RoboCup Middle Size League in 2003”, Robocup'2003 – Team Description Paper, Julho 2003,Pádua, Itália.

[4] Fernando Ribeiro, Ivo Moutinho, Pedro Silva, Carlos Fraga, Nino Pereira, “Vision, Kinematics and Game strategy in Multi-Robot Systems like MSL RoboCup”, ROBOCUP'2004 – Team Description Paper, Julho 2004.

[5] Philips Semiconductors (visit http://www.semiconductors.philips.com)

[6] 35) P. Lima, A. Bonarini, C. Machado, F. M. Marchese, C. Marques, F. Ribeiro, D. G. Sorrenti, “Omni-Directional Catadioptric Vision for Soccer Robots”, Special Issue of the Robotics and Autonomous Systems Journal, Elsevier, Amesterdão, Volume 36, nº 2, 31 de Agosto de 2001.