Design and Control of the Mobile Micro Robot Alice

Design and Control of the Mobile Micro Robot Alice

G. Caprari and R. Siegwart

Autonomous Systems Lab (ASL) , Institut d'ingénierie des systèmes (I2S)
Swiss Federal Institute of Technology Lausanne (EPFL)
CH–1015 Lausanne

E-mail: gilles.caprari@epfl.ch
http://asl.epfl.ch

Abstract. This paper describes the design methodology to build an autonomous mobile robot of just few cubic centimeters (2 x 2 x 2 cm). Starting from the expected features, the design path will be shown, going through solution proposition, component search, subsystems interconnections, several versions and ending to the final product. The second part of the paper will be dedicated to the control of such a robot which has implications in the control and software architecture. The overall design cycle results in the last version "Alice 2002" which has up to 10 hours autonomy with rechargeable battery, proximity sensors and remote control possibility. Furthermore many additional hardware modules like a radio or a linear camera can extend the basic capabilities. At the moment about 300 Alices exist and are used in various exhibitions and research programs.

1 Introduction

Mobile Micro Robots (MMR) are desirable for any small restricted place where a task should be fulfilled. In particular, where flexibility and adaptability is asked, MMRs may be the only viable solution. Examples could be exploration in small environments [8], fixing unexpected failures in small plants, process automation or micro factories. In parallel those small robots has been shown to be very convenient for basic research in collective robotics and bio-inspired robotics [1]. First because the small size permits to extensively use a large number of units in a reasonable space and second because the simplicity and the limited computational resources fit well with the relatively simple fundamental algorithms governing social insects. In return these researches should establish rules how to design the controllers of collaborating robots. A group of small simple robots might thus solve more complex tasks [2].

Our goal was thus to provide a MMR platform as small as possible, integrating the necessary elements to make an intelligent system. The description and the analysis of this process should also serve if possible as guideline for future design for other similar projects. Many implementation details depend on the starting expectations and the final application, however the basic aspects are to find in any development where miniaturization, system integration, system intelligence and autonomy play a central role. In such integrated products a typical characteristic is the interaction between hardware and software design and also between the composing subsystems [9]. In other words, all is interconnected and when you change something in the system it has often consequences also on other sections. It is thus crucial to have a good global overview and in a particular way the knowledge of all the relations within the system. Anyway this particular aspect of integrated systems often results in developing time longer than expected and some more prototypes and design cycles before a satisfactory end product.

Despite the mentioned interconnections, hardware and software will be presented in 2 separate sections trying to linearize the description of the robot whereas the reality of the development

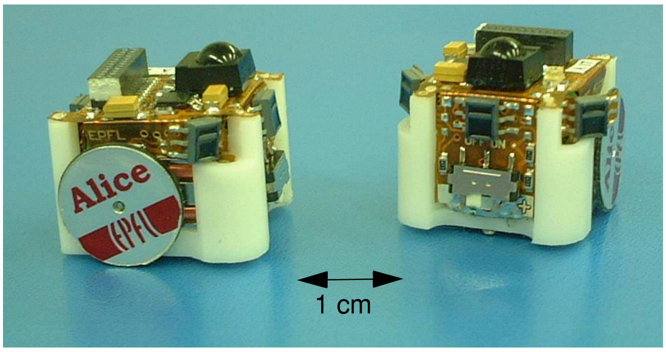

Figure 1. The last version Alice 2002.

went through several redefinitions which may probably just bother the reader.

2 Design of the mobile micro robot Alice

It will be presented here the last version of the robot "Alice 2002" (Figure 1). Previous versions were built starting from 1996 and improved over the years [7]. The first ideas were driven by the availability on the market of small microcontrollers in conjunction with low power motors. Further developments were pushed by new electronic components on the market or by new benchmark applications. Thus the goals are set by needs, ideas or dreams but in reality they are limited by the available technology and practical constraints.

There are 3 aspects that are fundamental in this research, the first 2 (size and mobility) come directly from the name Micro Mobile Robot, whereas the third aspect (intelligence) comes from the fact that robots have to deal autonomously as much as possible with the surrounding environment. Robots behave better when they handle by them self their task instead of being totally controlled by a remote intelligent unit. This is especially true for micro systems which have to operate quickly in spaces difficult to perceive by a distant controller. Another critical aspect that derives from the other 3 is the power management. As the robot is mobile it should have energy on board and because it is miniature and low weight it is almost impossible to supply energy with cables. The robot needs energy for the motors and for the onboard CPU in order to be mobile and intelligent but on the other hand the size constraint make difficult to carry a big powerful battery. The power supply should also stand for sufficient time, let's say at least half an hour up to several hours, otherwise the robot won't have enough time to fulfill its task.

Other factors to keep in mind before starting the implementation are simplicity, robustness, easy assembly and compactness which will influence the feasibility and afterwards the practicability of the product. Other details are center of gravity, ground clearance, location of the sensors, connectors position.

In the next sections the design process will be presented first setting the general requirements and than going through the key components before showing how they have been put together.

2.1 Requirements and expected features

The key expectation among the 3 fundamental ones mentioned before (size, mobility and intelligence) is for sure the small dimension. The robot should be as small as possible, that is, preferably smaller than any other comparable existing robot. There are quite a lot of mobile robots down to the size of 50-100 cm^3, many of them even made by enthusiastic hobbyists as nowadays small off-the-shelf components are available. Decreasing the volume it is possible but requires more time investment and integration effort. Reaching 10 cm^3 including motors, power, sensors and CPU seems to be today a kind of limit. Going further down it is certainly possible but at this point other techniques should be used. The actuators have to be fabricated out of a silicon die which also contains the controller circuit [6] and each subsystem is a research project itself. Our goal thus regarding miniaturization is to stay around the mentioned limit of 10 cm^3 which means also that the product is still manipulable and future extensions are reasonable to add.

On the mobility side, the robot should ideally be able to move as fast as possible on any terrain. In a first phase however it is sufficient to run on an almost flat surface at a reasonable speed otherwise the mechanics becomes too complex and control task too difficult. A reasonable speed for a robot is, among other parameters, dependent on the characteristic size of the robot itself. Typical values are around 5 times the size per second and this is more or less valid for big and for small autonomous robots. This gives a wished speed for our small vehicle of about 10cm/s.

The intelligence aspect is more difficult to define from the beginning and the requirements depends manly on the task. It will be shown later in the section about control that there are ways to extend the system intelligence even after constructing the robot (additional sensors, external supervisor, robot cooperation, ...). Anyway the basic robot should comprehend a sufficient powerfull CPU and at least some sensors for perception.

2.2 Components

The principal components of an autonomous mobile micro robot are actuators for locomotion, microcontroller, batteries, sensors and communication devices.

2.2.1 Actuators

As actuators responsible for the displacement we have chosen watch type motors (LAVET [3]) because they have a very low power consumption (0.5 - 2 mA), they are highly optimized (several years in the watch industry), easy to control (6 steps per rotor revolution and directly driven from 3 microcontroller pins) and a gear with ratio 180 is included in the motor block. This is probably the most critical component of the robot. There are no commercially available motor of this kind even if each quartz watch with moving hands contains at least one. We are using special bidirectional motors made by the Swatch group for the watch model T-touch sold by Tissot. The development of such a motor requires a big investment and depends on the final product so that there is no standalone standard motor module at the moment.

2.2.2 Microcontroller

The system is controlled by a PIC16F877 microcontroller from Microchip, which is also characterized by a low power consumption (about 1 mA @ 4MHz). Other important features are its size (PT44, 12x12x1 mm), the integrated peripherals (ADC, USART, ...) and the 8 Kwords of flash program memory. This allows the end user of the robot to change the control program within seconds according to its special needs, particular application or new extension hardware.

2.2.3 Power

Earlier version of the robot used 2 or 3 button cell also mainly used in the watch market. They fit perfectly to the robot dimensions but the problem with this kind of batteries is the low maximal current due to the high internal resistance. This limits the functionality and the extendibility of the system and thus we now use accumulators even if they have a much lower capacitance/volume ratio. The new solution also include a voltage regulator to stabilize the power supply and thus additional modules have enough power to work properly. Nevertheless when the system required much more milliamps, the voltage may decreased causing malfunctions to the motors and other electronic. For modules like the bidirectional radio, this may result in communications errors of a non-negligible extent. For such small systems power management remains a primary issue to deal with from hardware up to software.

2.2.4 Sensors

A robot should be equipped with as many sensors as necessary to perceive the surrounding environment and in order to solve its task. For a mobile robot this means at least be able to detect obstacles within a range that still permit to avoid them. Other sensors to measure some environmental parameters are very desirable but depends on the application. Thus indispensable are touch sensors, distance sensor or proximity sensors. At this size touch sensors may be quite fragile, precise distance sensors are difficult to achieve and thus only simple proximity sensors are feasible. Ultrasound or laser use the time of flight principle and for very short range would require highly sophisticated electronics. Active infrared proximity sensors are simple to use, inexpensive and can be found in compact packages. Figure 2 shows the sensor placement on the

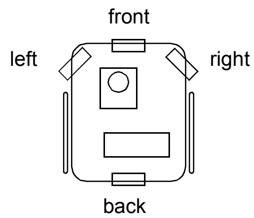

Figure 2. The 4 IR proximity sensors on Alice. 3 are placed in front and 1 in the back.

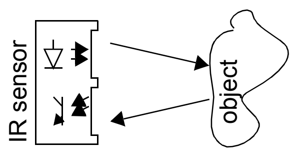

Figure 3. IR proximity sensor principle. An IR beam is emitted by the LED, reflected by the obstacle and measured by the photo diode.

robot and Figure 3 shows the well known working principle. Because the sensors are active, they also can be used as bidirectional communication devices. This is used for robot-robot local communication and it is an additional interesting feature for collective robotics. There are different IR proximity sensor and we decided to use the SFH9201 from Infineon mainly for the compact size and availability. The robot is able to see obstacles up to 4 cm away and to do local communication up to a distance of 6 cm.

2.3 Modularity

There are 2 main motivations for a modular hardware architecture when building such small robots. The first is intrinsically due to restricted size not allowing to put everything on one single robot. It is instead preferable to equip some robots with some sensors and other robots with some other functionality [4]. The second motivation is related to the partially unknown final application and also to the research still going on which both ask for future extendibility. It is therefore better to have an extension connector with as many as possible free lines and just the basic indispensable functionality on the base module. The advantage is that it is quite simple to add new features to the robot by just adapting the software and plugging on new modules. The disadvantage may be that adding many new modules (in our case on the top) tends to displace the center of gravity and thus the robot get mechanically unstable.

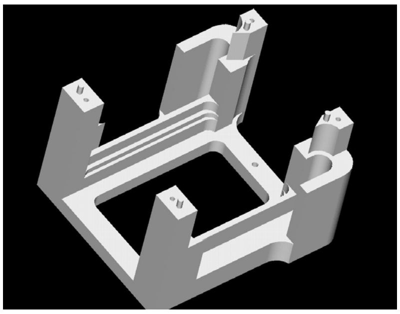 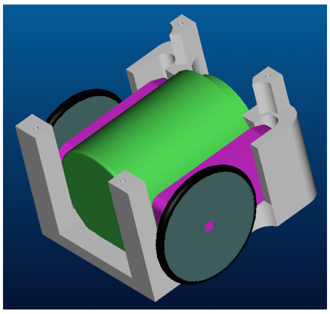

Figure 4. Basic robot mechanics. Left the empty plastic frame. Right the same frame with just the biggest components: 1 cylindrical rechargeable battery and 2 motors with wheels.

The modules developed and successfully used until now are: a radio module, an infrared receiving module, a linear camera and an higher range distance sensor. Simple modules for programming and debugging are also used and for the future a color camera and a gripper are under development.

2.3.1 The basic robot Alice 2002

The base of Alice 2002 (Figure 1) is a standalone mobile robot including all the necessary components (Table 1). These are: microcontroller, rechargeable battery, 2 watch motors with aluminum wheels and rubber tires, extension bus (connector), IR remote control receiver and 4 IR proximity sensors. Other secondary components are: resonator, power switch, voltage regulator, one LED and some capacitors and resistors.

Microcontroller	PIC16F877 (8 bit CPU, 6 RAM, 8KWord Flash)
Power supply	Varta 3/V40H, MiMH rechargeable
Actuators	2 bidirectional Swatch motors
Sensors	4 IR proximity sensors: SFH9201/1-2
IR receiver	TSOP1836ss3V

Table 1. Principal hardware components of the basic MMR Alice 2002.

This base is built as compact and robust as possible with a main emphasis on low cost production. The chosen watch motors have 4 electrical contacts that can directly be soldered onto a printed circuit board (PCB). They are mounted together with all the other electronic components on a flexible circuit board and bent in the required vertical position during the assembly process. Other electronic components remain in the horizontal plane whereas the PCB regions with the sensors are arranged around the chassis. The flexible print is mounted within a plastic frame (Figure 4) and the motors are kept in place by a special U-shape of the frame. This plastic frame holds also the battery and 4 holes permit to fix the extension modules on top of the robot. When all the components are fit in the right place, 4 points of the plastic frame are warmed up to hold the PCB and the frame together. No screw is used.

The design of the plastic frame is optimized for low cost production. It can either be manufactured by a milling machine all from only one side (top) or produced by plastic injection. Another advantage of this kind of plastic structure is that it is a little bit flexible and thus helps to protected against mechanical shocks. The other parts that deform in case of shock, are the rubber tires. The robot is thus to some extent protected from improper manipulations. The system and its functionality (Table 2) can be extended by just plugging on new modules on top of the base of the robot. The 24 pin bus connector has digital in- and outputs, analog inputs, power supply and a serial port.

Dimension	22 x 21 x 20 mm^3 (b,l,h)
Weight	11 g
Velocity	40 mm/s
Power consumption	12 - 18 mW
Power autonomy	up to 10hours
Proximity sensor range	40 mm
Communication	local IR 6 cm, IR remote 10m

Table 2. Characteristics of the basic robot Alice 2002.

2.3.2 Radio module

This module (Figure 5) contains the HX1000 device as transmitter and the RX1020 as receiver, both working at 433.92 MHz and made by RF Monolithics, inc (www.rfm.com). With this transceiver the robot is able to communicate via a radio link with other robot or with an host computer. This feature is of high interest to exchange information between robots even if they don't see each other or to supervise the status of a robot by means of an host computer. The transmission of data in both directions was tested up to a distance of about 10 meters at 1 kBaud.

2.3.3 Infrared receiving modules

The base of Alice 2002 already contains an IR receiver for standard TV remote controllers. This allows to control the robot with a cheap, easy to find device. The communication is unidirectional, from the user to the robot and the feedback is visual. For extremely low power applications and for former versions of Alice, a cheap infrared receiver module with low power consumption (0.2 mA) has been built. The components of this unit are a photodiode, an amplifier and a filter. With a simple infrared remote control it is then possible to guide the micro robot or to send parameters.

2.3.4 Linear camera

To be able to perceive features far away, the robot can be equipped with a linear camera. We used the sensor TSL2301 by TAOS and we created the optics around it. This results in a module to be plugged on top of Alice (Figure 6), a little bit smaller than the robot itself and can be read out directly by the main processor with a serial protocol. The chip has 102x1 pixels and an integrated 8 bit ADC so that the PIC can easily manage the information flow. It still works at 3

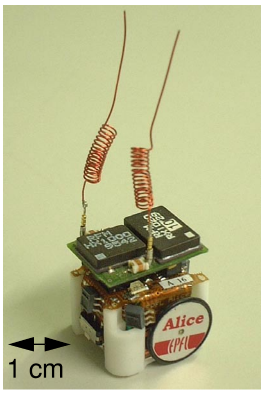

Figure 5. Alice with the radio module.

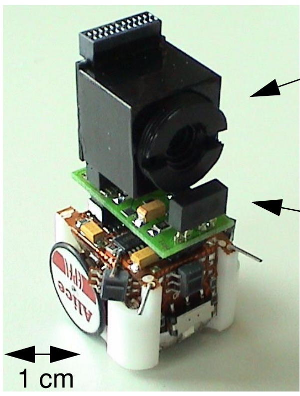

Figure 6. With an additional distance module and the linear camera.

Volt and with a picture refresh rate of 50 Hz, the consumption is 2.2 mA. The typical features that such sensor can detect are bright spots, vertical black and white lines and the width of these.

3 Control software and algorithms

There are different ways to control robots, ranging from totaly remote controlled to completely autonomous. The control method proposed in somewhere in between.

The robot has some pre-programmed basic behaviors and communications abilities. Using this base, the control architecture put together an intelligent general behavior depending on the task. The underlying software architecture with the operating system permits the pseudo-parallel execution of all the subtasks. The next sections will explain this concept starting from the description of the implemented behaviors and the communication abilities through the general control architecture until the real implementation of the simple and compact operating system.

3.1 Behaviors

For the pre-programmed basic behaviors like obstacle avoidance and wall following, Alice uses its 4 infrared proximity sensors for environment perception (Figure 2). The sensors are active (Figure 3) and controlled by the software in the on-board microcontroller PIC16F877. The procedure is as follows:

- the PIC first reads the analog value on the IR phototransistor, converts to a digital value and this is the ambient light,
- then it turns on the IR diode that beams out light to support the measurement,
- after a stabilization time (~200 ms) a new analog value is read, the IR diode is turned off again, the value is subtracted from the ambient light and this difference is the light reflected by any visible object in front of the sensor

The proximity measures are then used for obstacle avoidance, wall following or other behaviors acting finally on the motors speed (Figure 7).

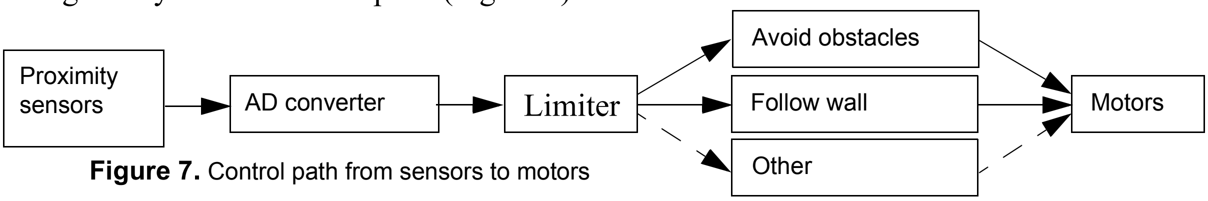

Figure 7. Control path from sensors to motors

The sensor response is not linear with the distance and depends also on the color, surface quality and the angle of the object to detect. Further, the sensitivity region is of conic shape with a quite big opening angle of about 60 degrees. In order to avoid the disadvantages of the high non

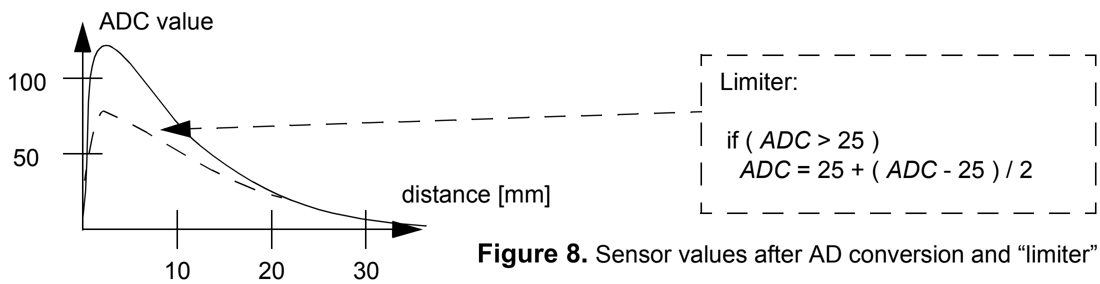

Figure 8. Sensor values after AD conversion and “limiter”

linearity, a very simple “limiter” (Figure 8) is introduced in software which helps to get something more linear.

3.1.1 Avoid obstacles

Obstacle avoidance is made with a PD (Proportional-Differential) controller. The differential part is just the difference between the recent sensor measure and the former one. As the controller is implemented in assembler on a 8 bit microcontroller it is preferred that the coefficients are integer. In fact coefficients equal 1 for both *P* and *D* turned out to be good enough. Next the formulas are given:

```
Motor_Pause = P * ( distRight - distLeft ) + D * ( ( distRight - oldRight ) - ( distLeft - oldLeft ) )     (1)
with  P=D=1  :  Motor_Pause = 2 * distRight + oldLeft - ( 2 * distLeft + oldRight)                           (2)
```

If the sensor value is too low, that is, near to noise, no direction change is done. Otherwise one motor is slowed down whereas the other just run at max speed with (see *(3)* to *(9)*).

```
if  ( distRight > noise OR distLeft > noise OR distFront > noise)  // just noise?                            (3)
        if ( Motor_Pause > 0 )                                                                             (4)
                Mleft_Pause = Motor_Pause + distFront ;  Mright_Pause = 0 ;  // turn left                  (5)
        else                                                                                               (6)
                Mright_Pause = ( - Motor_Pause) + distFront ;  Mleft_Pause = 0 ; // turn right             (7)
else                                                                                                       (8)
        Mright_Pause = 0 ;  Mleft_Pause = 0 ;  // go straight                                              (9)
```

Acting on the motor pause instead of directly on the motor speed is particular to this kind of watch motors (see "Alice motor control" on the Alice doc web page [5]) so the higher the *Motor_Pause* the slower is the motor: *Motor_Speed ~ 1 / (30 + Motor_Pause)*

In order to have more effect on the direction change and thus the avoidance of the obstacles, not only the motor is slowed down but from a certain value it turns in the opposite direction (see *(10)* to *(14)*). This result in an increased dynamic of the robot movement.

```
if  Motor_Pause > 20                                                                                       (10)
        Motor_Pause = 40 - Motor_Pause ;                                                                   (11)
        direction = -1 ;  // invert direction and run backward this motor                                  (12)
        if  Motor_Pause < 0                                                                                (13)
                Motor_Pause = 0 ;  // max speed backward                                                   (14)
```

3.1.2 Follow objects

Another interesting behavior is follow or find objects. This is simply done by switching right and left in equation *(1).* As soon as the robot detects an object, it will be attracted by this and it will stay as near as possible. If the object moves, the robot will follow it as is the case of another robot moving in front. A chain of 5 following robots was demonstrated.

3.1.3 Follow walls

The robot is also able to follow walls on its right or left controlling the distance between the respective side sensor and the wall or object. For this it first goes straight looking for a wall and then tracks it with a PD controller similar to the obstacle avoidance. For the case of follow right, the next pseudo code lines present the algorithm:

```
Motor_Pause = P * ( distRight - FollowDistance ) + D * ( distRight - oldRight )                            (15)
implemented :  Motor_Pause = 2 * distRight - ( 35 + oldRight )                                             (16)
if  ( Motor_Pause > 0 )                                                                                    (17)
        Mleft_Pause = Motor_Pause  ;  Mright_Pause = 0 ;  // turn left                                     (18)
else                                                                                                       (19)
        Mright_Pause = ( - Motor_Pause)  ;  Mleft_Pause = 0 ; // turn right                                (20)
```

3.2 Communication abilities

As already mentioned in the previous sections presenting the hardware, the robot has 3 main

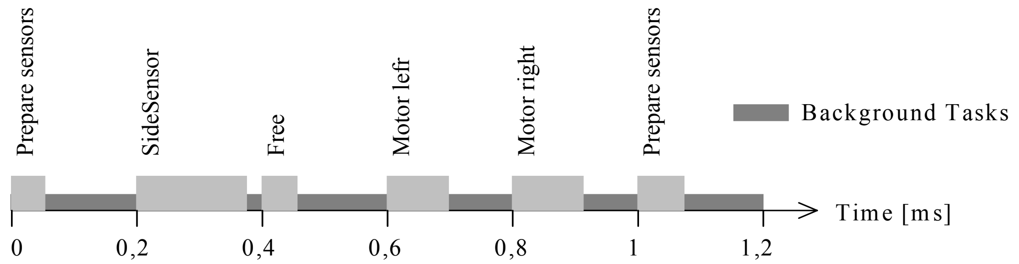

Figure 9. Pseudo parallel task handling. A scheduler call every 200 us the next task. In background there is time for longer time uncritical jobs.

communication methods. Depending on the application the simpler one should be chosen.

3.2.1 User-robot one way remote control

With a simple standard infrared TV remote control it is possible to send commands to the robot like forward, turn left, backwards and so on. An extension to this is when instead of a human, it is a computer that automatically take the control. In order to close the control loop the PC should have a camera to check the scene.

3.2.2 PC-robot bidirectional communication

This method uses the radio module and was better presented in [8]. One use is gathering sensor data from the robot and navigate it via a PC-user interface. With regards to energy t, this is the better way to use when the communication has to be bidirectional and when the required communication range exceed 20-30 cm.

3.2.3 Local communication

For collective robotics and for scalable systems, it is of high interest to have a communication which is only local. For this the IR fits well and the already on-board proximity sensors make it possible. By just using another frequency for distance and for communication, simple messages can be exchanged between nearby robots. Furthermore as Alice have 4 independent IR sensors, it knows from where the signal is coming in and also which one of the 4 sensors is used by the other robot. Of course the quality of the communication is somehow restricted, particularly in case of bad alignment, multiple robots emitting at the same time in the same place or when the robots are too far away (> 6 cm). The control algorithm has to take into account this but we could show nice results where a group of robots was able to take simple decisions in common.

3.3 Control architectures

When the robot is controlled by a human or by a remote computer, we speak of multi stages control, because in any case there will be a first control loop on the robot itself, then a second through an host PC via radio or vision, and finally a third one involving the global task or an operator. On the other hand when the robots are in a colony we speak of social collective behaviors. The emerging global behavior depends on the rules in each single unit but also particularly on the relations and reactions to each other. Those may be governed by inexplicit or explicit communication at local level. The robot needs these different approaches but is at the same time a good test bed for exploring and optimizing those control architectures.

3.4 Software architecture

Because of the PIC limitations and in order to optimize all the system, it make sense to program the robot in assembler language. The instruction set is limited at 35 instruction (RISC) and the code is hardly readable. For this reason a mix assembler-C is proposed, keeping the low-level part in assembler and some middle-high level algorithms in C language. The code written in assembler is in charge of controlling the 2 motors, reading the 4 proximity sensors, performing local IR communication, decode the incoming the IR commandos and also provides a set of simple behaviors like obstacle avoidance or wall following. On the other side the C code, which is more readable and easier to write, is supposed to control the general behavior of the robot using the underlying structure. For example the robot could run around avoiding obstacles and as soon it detects another robot it perform a “dance” and exchange a simple message. The assembler part is organized as a simplified multitask real-time operating system. Basically every 200 us an interrupt routine is called and inside this a scheduler runs every time the next task (Figure 9). There are 5 different tasks so that every task is called at an interval of 1 ms. This jobs can be further split or can have longer cycle composed by several subtasks. This is the case for example with the sensor reading, where the whole cycle takes about 50 ms for first reading the ambient light and the proximity on each of the 4 sensors and afterwards the same sensors are used for local communication. During the idle time, when none of the major five tasks is processing, some background jobs can be done and it is at this point where the C-algorithm runs. As this is intended to react to ambient changes (sensors) or to communications ant to take decisions.

4 Conclusion and outlook

The micro mobile robot Alice was presented in particular regarding components, hardware design, control architecture and capabilities. The development path was introduced, starting from the ideas and the requirements, to the appropriate components and mechanical realization until the control and software structure. The taken design decisions and its discussion should help for similar developments. The robot itself is very compact and optimized but extendible and thus potentially utile even for projects we never thought about. The gained knowledge and the result achieved let hope for a great future of this and other MMRs.

References

[1] **F. Mondada, E. Franzi, P. Ienne**, Mobile robot miniaturization: A tool for investigation in control algorithms, *Proc. of the 3rd International Sym. On Experimental Robotics*, pp. 501-513, 1993

[2] **E. Bonabeau, M. Dorigo, G. Theraulaz**, Swarm Intelligence: From Natural to Artificial Systems, *Oxford University Press*, July 1999.

[3] **J.-C. Bernety**, Mype LAVET, *Bulletin Annuel de la SSC, Vol. X*, 1981.

[4] **L.E. Navarro-Serment** et al., Modularity in Small Distributed Robots, *SPIE vol. 3839*, pp. 297-306, 1999.

[5] **Alice 2002 doc page**, *http://asl.epfl.ch/research/systems/Alice/alice02Doc.php*

[6] **R. Yeh, K.S.J. Pister**, Design of Low-Power Articulated Microrobots, *Proc. International Conference on Robotics and Automation, Workshop on Mobile Micro-Robots*, April 23-28, 2000, pp. 21-28.

[7] **G. Caprari, K.O. Arras, R. Siegwart**, The Autonomous Miniature Robot Alice: from Prototypes to Applications, *IROS'00*, pp 793-8, 2000.

[8] **G. Caprari, K.O. Arras, R. Siegwart,** Robot Navigation in Centimeter Range Labyrinths, *5th International Heinz Nixdorf Symp., Autonomous Minirobots for Research and Edutainment*, pp 83-92, Germany, 2001.

[9] **G. Caprari, T. Estier, R. Siegwart,** Fascination of Down Scaling - Alice the Sugar Cube Robot, *Journal of Micromechatronics*, VSP, Vol 1, No 3, pp 177-189, 2002.