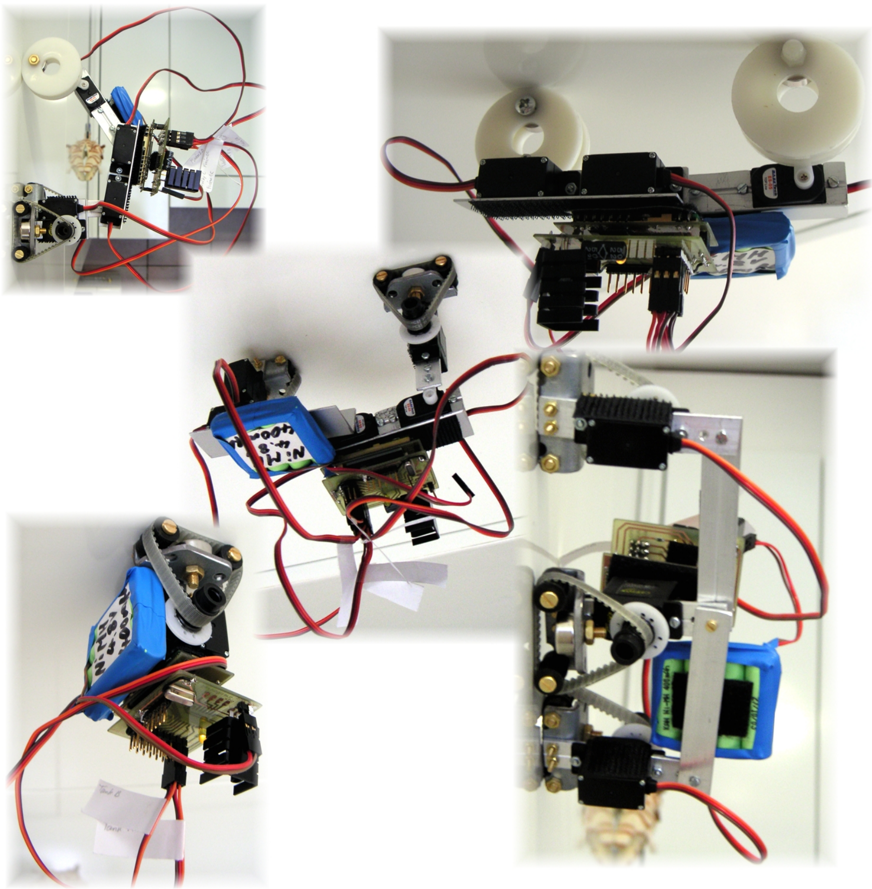
LSRO1
Laboratory of Robotics Systems 1

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Miniature mobile robot for inspection of ferromagnetic structures
MagBot

Pascal Gilbert
June 21, 2007



Professor Hannes Bleuler

Assistant Frédéric Rochat
Assistant Dr. Francesco Mondada

LABORATOIRE DE SYSTEMES ROBOTIQUES 1
CH - ES ROBOTIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

PROJET DE SEMESTRE – ETE 2007

Titre:	Miniature mobile robot for inspection of ferromagnetic structures		
Candidat(s):	Pascal Gilbert	**Section:**	MT
Professeur:	Hannes Bleuler		
Assistant 1:	Frédéric Rochat	**Assistant 2:**	Francesco Mondada

Donnée & travail demandé :

There is a growing interest for the development of miniature mobile robots for the inspection of complex infrastructures, such as turbines, small pipes or valves. The LSRO1 is involved in several projects for the development of this type of devices, some of them aiming at exploring ferromagnetic structures. Previous semester projects have already studied an interesting concept of inspection robot for ferromagnetic environments. This robot has a size of about one cubic dm.

The goal of the project is to develop a smaller mobile robot (size about 100 x 50 x 50 mm) able to move on vertical magnetic structures. The project includes the mechanical design, the adaptation of existing electronic circuits for the control and some software to remotely control the robot using, for instance, a bluetooth connection.

Remarques :

Un plan de travail sera établi et présenté aux assistants avant 23 mars 2007.

Une présentation intermédiaire (environ 10 minutes de présentation et 10 minutes de discussion) de votre travail aura lieu dans le courant du mois de mai 2007. Elle a pour objectifs de donner un rapide résumé du travail déjà effectué, de proposer un plan précis pour la suite du projet et d'en discuter les options principales.

Un rapport, comprenant en son début l'énoncé du travail (présent document), suivi d'un résumé d'une page (selon canevas), sera remis le lundi 25 juin 2007 avant 12 heures en 3 exemplaires à l'assistant responsable. L'accent sera mis sur les expériences et les résultats obtenus. Le public cible est de type ingénieur EPF sans connaissance pointue du domaine. Une version préliminaire du rapport sera remise à l'assistant le 18 juin 2007. Tous les documents en version informatique, y compris le rapport (en version source et en version pdf), le document de la présentation orale et un résumé au format pdf, ainsi que les sources des différents programmes doivent être gravé sur un CD-ROM et remis à l'assistant au plus tard lors de la défense finale.

Une défense de 30 minutes (environ 20 minutes de présentation et démonstration, plus 10 minutes de réponses aux questions) aura lieu dans la période du 26 juin au 5 juillet 2007.

Le professeur responsable:	L'assistant responsable:
Signature :	Signature :
Hannes Bleuler	Frééric Rochat

Lausanne, le 12 mars 2007

Miniaturized mobile robot for inspection of ferromagnetic structures

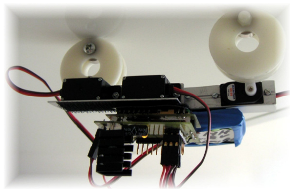
Pascal Gilbert, Section Microtechnique

Assistants: *Dr. Francesco Mondada, Frédéric Rochat*

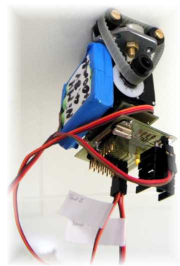
Professor: *Hannes Bleuler*

Overview

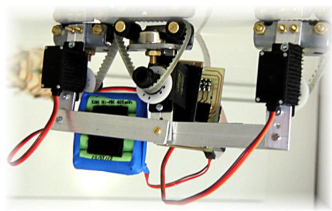
This project aims to develop a *miniaturized mobile robot for inspection of ferromagnetic structures*. The robot should be small enough to fit inside water pipes with a diameter of typically 1 inch (2.5 cm), for inspection purposes. However, if a mechanism is found which has the potential to be miniaturized, this constraint does not necessarily need to be fulfilled in a first stage. Besides piping, tanks need to be inspected as well. They are composed of several segments which are soldered together, forming a structure of all kinds of perpendicularly intersecting surfaces. The robot needs to be capable to overcome these obstructions.



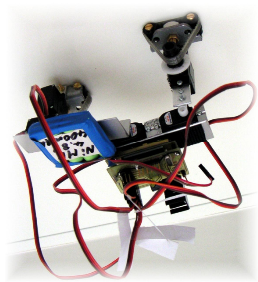
walky



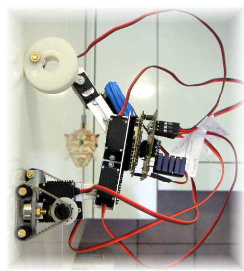
tank



platoon



fusion one



fusion two

Prototypes

After carefully evaluating a long, but not exhaustive list of solutions several prototypes have been built and tested. Basically two kinds of locomotion have been used, coasting and walking/sliding. In a final phase, combinations of the two techniques have been attempted.

By putting together highly modular parts, several different variants could be tested rapidly. The family counts five individuals: *walky*, *tank*, *platoon*, *fusion one*, and *fusion two*.

A wireless link connects the robots to a computer, from where the servo motors may be controlled. Optionally a joystick may serve to pilot the robot directly. It takes some practice to handle all the degrees of freedom of the more complex models such as the *fusion one* or the *fusion two*.

Results

While excellent solutions have been found to move on surfaces with any orientation (normal, upside down, vertical), showing a high mobility, corners remain somewhat of a problem. *fusion one* shows promising behavior in overcoming them, but it still needs some perfection. Keeping in mind that all these prototypes are of first generation, the overall results are quite satisfactory.

The most performing robots are able to take on a payload of about half a kilogram, they are able to turn on the spot and have a top speed of about 23 $^{mm}/_{sec}$.

To be continued …

There is a lot of potential in some of the tested solutions. Developing further will soon yield a robot which answers the entire specifications, which is small enough to fit inside the pipes to be inspected, and slick enough to overcome obstacles typically found in its field of application. Modularity allows to test other combinations as well, or to extend the set of modules.

Contents

Chapter 1

Introduction

This project aims to continue the research for a reliable solution to displace a robot in a ferromagnetic environment [2] [4]. The previously elaborated solution catalog has been revised as well as new, more innovative solutions have been developed. Main goal is to build and test two functional prototypes, which may not exactly meet the specifications or may have limited mobility, however, they are very well adapted to be miniaturized.

Several attempts have been made to develop robots which are able to climb up ferromagnetic structures, usually with a very well defined purpose such as labeling oil tanks [10], climbing a wall [8], or inspecting pipes [11].

First an entire catalog of solutions will be presented. *Chapter 2 Solution Catalog* summarizes the brainstorming process which has been taking up a considerable amount of the project's time frame. A great deal of the basic concept is already described, which will help to understand the hows and whys of the different prototypes. *Chapter 3 Modularity* then presents the *primitive* modules used to build the different robots. It has been tried to keep the entire project as modular as possible giving the advantage to try out new ideas quickly. Core piece of the present work are, without a doubt, the prototypes, which are presented in detail in *Chapter 4 Prototypes*. It is recommended to make use of the enclosed CD, where more pictures and video clips may be found, as well as the entire source codes, software, reference papers, and so on. Ideally this project paper should be read while sitting in front of the prototypes in order to comprehend their functionality more easily. Finally, *Chapter 5 Ideas* briefly presents two more thoughts, which could not have been realized in the given time frame. The work is rounded off with *Chapter 6 Conclusion*.

Chapter 2

Solution Catalog

This Chapter aims to summarize the brainstorming done while searching for new solutions to the given problem. Each solution is briefly presented in form of a description and a schematic drawing as visual aid.

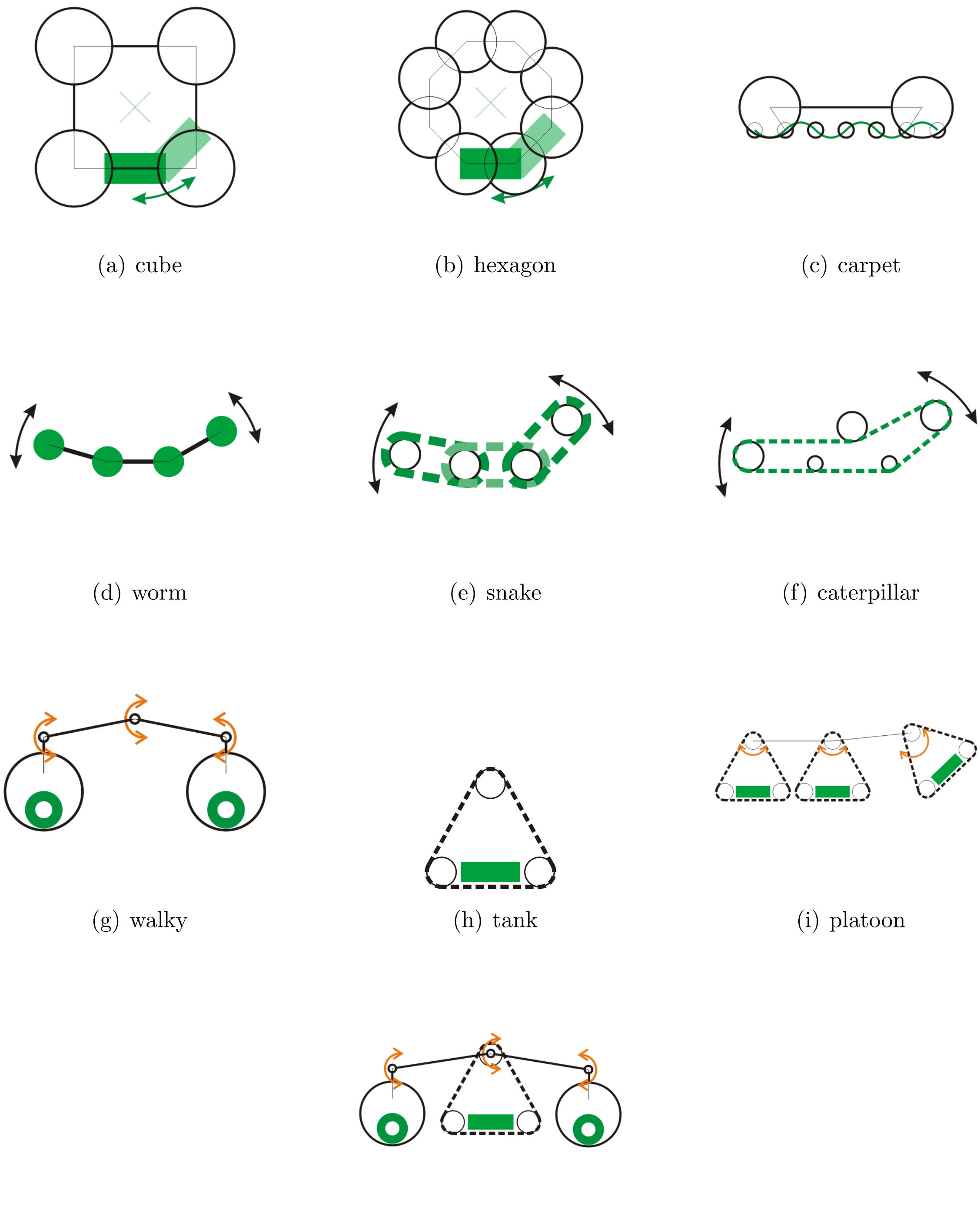
2.1 The Collection

The green colored elements represent permanent magnets. Coasting mechanisms as well as walking mechanisms have been studied. Hopefully it will be possible to combine the advantages of the two yielding the most performing robot.

cube 2.1(a) Based on the previously realized idea by *Christophe Groux* [4], another type of segmented-wheel robot has been studied. Instead of a purely passive mechanism to move the magnets toward the new surface, actively orientable magnets are used. The pilot is able to move two magnetic bars independently of each other to any desired direction, whereas one of the bars is capable of holding the entire robot upside down. When bumping against a new surface, one magnet may be oriented appropriately for the transition to take place. The eight wheel base allows the robot to continue on any one of the four cube faces, thus the robot does not necessarily have to change its orientation in space.

hexagon 2.1(b) While developing the idea of the *cube*, it was soon apparent that the transitions between surfaces of any kind of angle can be facilitated by extending the base to a hexagonal shape. The principle with the two orientable magnetic bars remains the same. Apart from the fact, that these two first structures present multiple wheels as contact points, the principle remains closely related to the one already realized by the preceeding projects [2] [4].

carpet 2.1(c) A rather new approach to the problem presents the magnetic *carpet*. A wheeled robot base is trailing a carpet-like structure dead underneath it, which is composed of several small magnetic components molded into an elastic cloth. Narrowly spaced, small supporting wheels prevent the structure to get stuck on the ferromagnetic underground, but still allowing the carpet to adjust to all kinds of shapes. Changing surfaces with a sharp angle remains a critical phase, since the carped would need to be plied in order to make the transition.



(a) cube (b) hexagon (c) carpet

(d) worm (e) snake (f) caterpillar

(g) walky (h) tank (i) platoon

(j) fusion

Figure 2.1: solutions

worm 2.1(d) The *worm* is composed of magnetic wheels. Several of these medium sized wheels are hooked together with freely rotating joints. The wheels have to be fine-tuned in such a way, that the attractive force is not too strong in order to facilitate the transitions of perpendicular angles. Several wheels will be needed in contact with the surface to maintain the vehicle upside down.

snake 2.1(e) Extending the magnetic wheel idea to a magnetic caterpillar resulted in the *snake*. Several caterpillar segments are linked together, again by freely rotating joints. As pretty as this structure might seem, there is a fundamental flaw which renders the idea completely useless. If the segments have to transit between angled surfaces, the entire contact area needs to be detached. Worst case again being a 90° angle, the caterpillar segment detaches almost parallel to its attraction force, resulting in an extremely high torque on the motors.

caterpillar 2.1(f) Combining some of the above ideas, the single *caterpillar* with adjustable ground-contact shape emerged. Thanks to the two rotary joints, as well as a lightly clamped caterpillar (doted with small permanent magnets) allows the structure to smoothly transition between angled surfaces. The caterpillar does not need to detach lengthy parts at once. Thought the mechanical structure is slightly more complicated, this solution seems somewhat promising.

walky 2.1(g) A completely different approach examines walking mechanisms. As it can be seen in nature, biped, quadruped, six-, or eight-legged creatures seem to be extremely well adapted in overcoming obstacles and rough terrain. However, walking robots remain a technical challenge in their realization, as they usually possess several highly unstable configurations. Taking advantage of the ferromagnetic underlying structure, the feet may exert a magnetic force, helping the mechanism to maintain a stable position. As prototype for proof of concept, a biped, the *walky*, is suggested for evaluation.

tank 2.1(h) Trying to combine the ideas of walking and coasting, the suggested magnetic feet of the *walky* may be replaced by small caterpillar like structures. They are composed of a fixed permanent magnet directed toward the surface of contact, and a rubber caterpillar, transmitting the rotary movement to the ground. This rather small module may be used as is, or in combination with a second, identical one to form some sort of *tank* with differential drive.

platoon 2.1(i) Developing the *tank* idea even further, the fixed caterpillar elements could be chained together by freely rotating joints, coming back to the idea of the *snake* idea. Although the *platoon* links the caterpillar elements more independently together, freeing some of the constraints of the *snake* by powering each segment separatly.

fusion 2.1(j) If the above concept prove to work nicely, the advantages of the *tank* and the ones of the *walky* may be combined to form the *fusion* robot. On smooth and plane surfaces, the legs can be kept off the ground, allowing the caterpillars for a fast displacement. Is there an obstacle to overcome, the legs can be activated, allowing the robot to elegantly walk past whatever is keeping the caterpillars from advancing.

2.2 Selection

Based on the above assessment, it has been decided to build some functional prototypes. It is of particular interest to evaluate the *caterpillar* (Figure 2.1(f)) and the *walky* (Figure

2.1(g)) more closely. By constructing three caterpillar modules as they are used in the *tank* (Figure 2.1(h) several combinations may easily be tested, such as the *platoon* (Figure 2.1(i)) or different configurations of the *fusion* (Figure 2.1(j)). The different parts are built as modular as possible to increase prototyping efficiency.

A fundamental difference between the *caterpillar* structure and the other robots is the way in which the permanent magnets are displaced. While the *caterpillar* maintains the magnets stationary in respect to the ground once they made contact, the other robots move the magnets in extremely close proximity to the underlying structure. *Lenz's Law* [5] stipulates that any action caused by electromagnetic phenomena results in a reaction countering its cause. In other words, the robot will encounter an opposing force as it moves its magnets forward. The closer the magnets are to the ground and the faster they move with respect to the ground, the bigger this opposing force will be. A detailed elaboration of this problem has been carried out by *Vincent Chenal* [2].

Chapter 3

Modularity

Inspired by the well known toy *LEGO*®, the developed prototypes (presented in the next Chapter) are based on small, simple modules which can be combined in any imaginable way. Of course the set is quite limited concerning the linking elements, however with crafting simple aluminum profiles, possibilities seem endless to come up with new robots. The present Chapter will present briefly the three major modules which have been used throughout the project. Any details in how to assemble the robots, and what additional parts are needed, are to be found in Chapter 4 on page 13.

3.1 Mechanical Elements

Basically three parts have been developed. The first structure is a *backbone* with a single 180° joint in the middle. At each extremity, another 180° joint is mounted which may be equipped by feet or other modules.

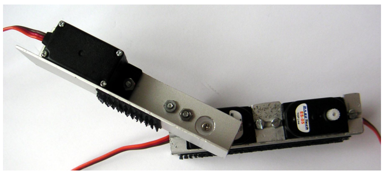
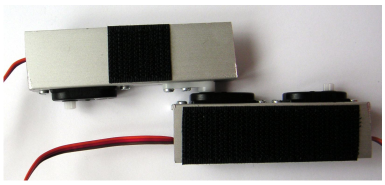



Figure 3.1: backbone

Next, two very simple *magnetic feet* have been constructed. The design allows to lift off the foot without any significant effort by turning it first, thus rotating the eccentrically mounted permanent magnet away from the surface. The feet mount directly onto the joints of the *backbone*.

The most complex module is the *fixed caterpillar*. A caterpillar is suspended on three supporting wheels, whereas the top center wheel is powered by a modified servo (refer to Appendix A.1 on page 29). A permanent magnet is located between the other two supporting wheels on the inner side of the caterpillar. Its magnetic flux is brought back to the surface by an L-shape ferromagnetic support, which may be seen as chassis of the

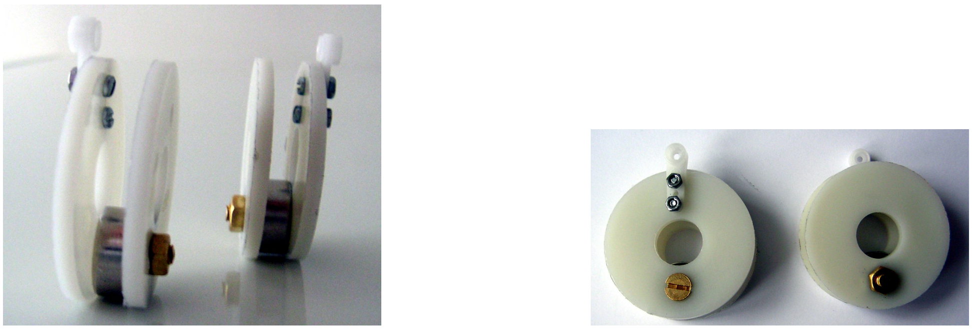


Figure 3.2: magnetic feet

entire module.

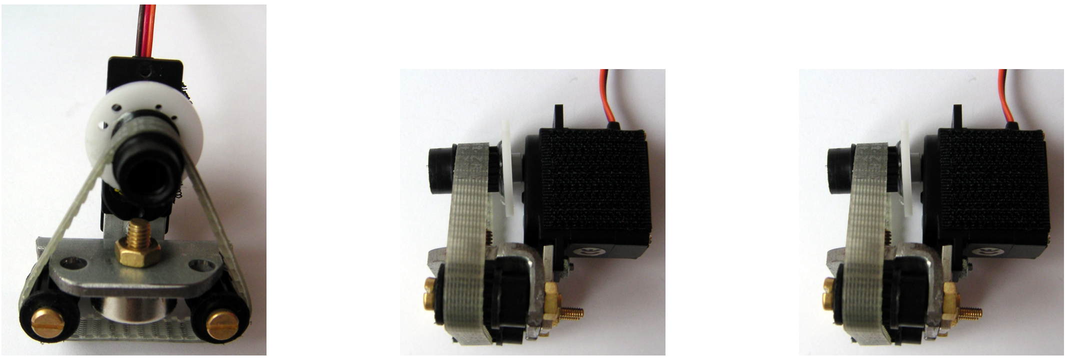
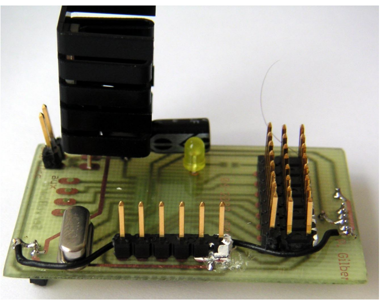


Figure 3.3: fixed caterpillar

3.2 Electronics



electronics	19 g
battery	44 g

Figure 3.4: electronics

Since all the used actuators are based on standard servos (even the ones used for continuous rotation, see Appendix A.1 on page 29), a simple servo controller has been developed. This module interfaces through UART to a ready-to-use radio tranciever (*EasyRadio* [6]), which then links the controller to a computer running a control applet (refer to Appendix B.2 on page 37) and equipped with a joystick.

Figure 3.5: easyRADIO module

Core piece is the *ATMEL ATtiny2313* ([1]) microcontroller which has been programmed to serve up to eight servos by using just two timers (refer to Appendix B.1 on page 31). This programming technique allows to reduce processor load considerably. Commands transmitted from the computer allow to set a certain position (which corresponds to a certain speed and direction for the modified servos), as well as the activation/deactivation of each servo individually. The latter feature proves to be quite handy to make a modified servo stop. Entire positioning sequences may be programmed into the Visual C++ applet used to control the robot. The connected joystick may be used to control each servo manually.

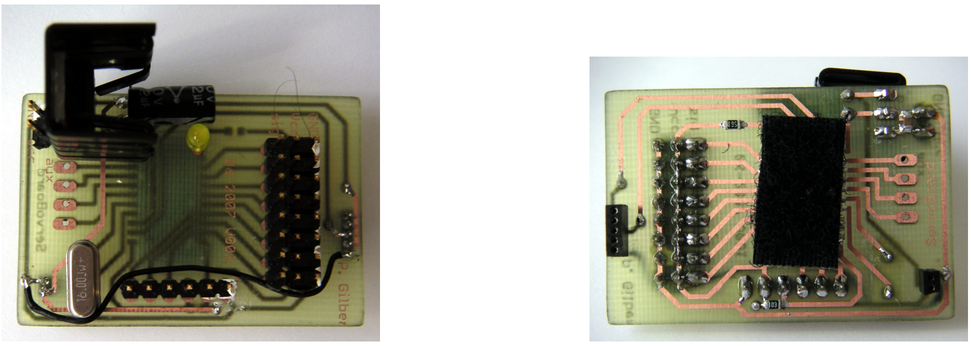


Figure 3.6: servo controller board

The servos need necessarily be supplied with at least 5 volts, whereas the microcontroller and the radio module do not support more than 5.5 volts. For this reason a linear low-dropout regulator has been implemented, which should be connected to a battery up to about 15 volts. The system works fine with an NiMH battery of 4.8 volts, which in practice shows a little more than 5 volts when fully charged. Once the charge drops below about 4.8 volts, the system starts to behave strangely, indicating that the battery needs recharging.

For details on the electronic boards please refer to Appendix C.1 and C.2 on pages 38 and 40 respectively. The controller's source code may be found in Appendix B.1 on page 31.

Chapter 4

Prototypes

As it proves to be extremely hard - one might even say impossible - to come up with a fairly accurate magnetic model describing the interactions between permanent magnets (or coils) and ferromagnetic structures, this project will mainly be based on experiments and the vast know-how of people working in this domain. Moreover, it is not perfectly clear under which conditions the robot will have to work, there is a big difference if the permanent magnet is to be placed on an extremely thin metal plate (some millimeters at the most), where flux saturation limits the force, or on a solid structure, where the underlying ferromagnetic material can be seen as infinitely thick from the point of view of the magnetic flux. Hence several prototypes have been built and tested. These machines are of course not to be seen as final products, since rather rudimentary approaches have been implemented for fast prototyping and evaluating the concepts. The mechanisms are presented in chronological order, letting the reader get a feeling of the evolution of the project.

Each robot has been undergoing the same tests to compare its performances. To measure speed, the covering of a distance of 500 mm has been timed on an upside down surface. Choice of this kind of surface was intentional, since certain prototypes could not reach there absolute maximal speed while inverted.

The payload has been measured without any electronics or batteries attached to the structure. The robot has been placed on an upside down metallic plate, about 3 mm thick with a regular hole pattern of 3 mm holes, spaced about 10 mm. This test plate is far from being optimal from the point of view of the magnetic flux, thus yielding results closer to reality.

The robot's weight is again measured without any electronic components or batteries mounted. The weights of the electrical parts can be found in Section 3.1 on page 10.

Corners of the test environment are to be seen as two surfaces intersecting at 90°. A convex corner designates an "outside" corner, whereas concave means it is seen from the "inside".

4.1 Caterpillar

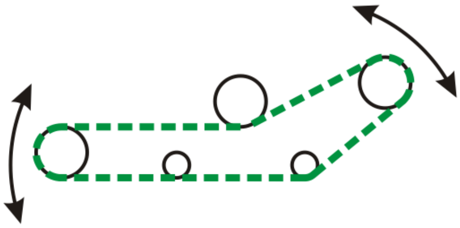


Figure 4.1: caterpillar schematics

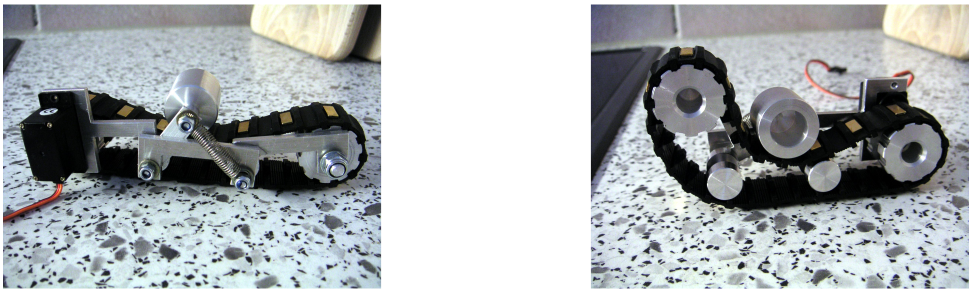


Figure 4.2: *caterpillar* prototype

Trying to realize a structure which adapts itself to any shape of the terrain, the idea of the flexible caterpillar emerged. Equipped with two joints and a clamping wheel, the caterpillar can adjust its shape to overcome any joints, corners or other kind of obstacle. Quickly building a prototype with *LEGO*® raised hope of having found an interesting structure. Being under the constant pressure of time, some technical drawings have been made and the prototype was being built. The parts were crafted in the LSRO's workshop for students.

4.1.1 Mechanics

The caterpillar out of the *LEGO*® box is being reused for evaluation purposes. Small permanent magnets [9] have been glued on the caterpillar. For the final product, it would be possible to mold these magnets directly into the polymer of the caterpillar.

Since the robot should influence the magnetic field as little as possible, aluminum has been used for the supporting structure and the wheels.

4.1.2 Electronics and Software

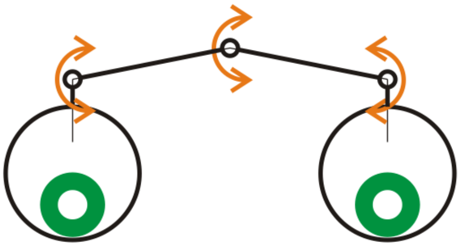
One single servo modified for continuous turning (refer to Appendix A.1 on page 29) has been used to power the caterpillar. The C++ applet (refer to Appendix B.2 on page 37) allows to control the speed and direction by interacting with the corresponding slider. Refer to Section 3.2 on page 11 for details about the controller.

4.1.3 Testing and Results

The construction of the robot already turned out to be more challenging than initially estimated. Complexity of mechanical parts usually renders them sensible to malfunctioning and loss of performance. The finished prototype seems to confirm this assessment, as it behaved in ways which have not been foreseen while planning the *caterpillar*. Too many degrees of freedom led to bending and turning of the structure without any advancement at all. The adding of springs and mechanical stops would be needed to complete the prototype to function properly. However, by adding spring-loaded joints, the flexibility of the mechanism would be compromised. Furthermore the structure would yet become more complicated, leaving little hope of ever achieving any satisfactory result.

Furthermore the magnets glued to the plastic caterpillar did not produce enough force to maintain the robot on an upside down surface. After carefully evaluating the behavior of this first prototype, the idea was abandoned.

4.2 Walky



normal surface:	OK
vertical surface:	FAIL
upside down surface:	OK
corners:	not tried
speed:	7.25 $^{mm}/_{sec}$
mobility:	straight line
weight:	70 g
payload:	670 g

Figure 4.3: walky

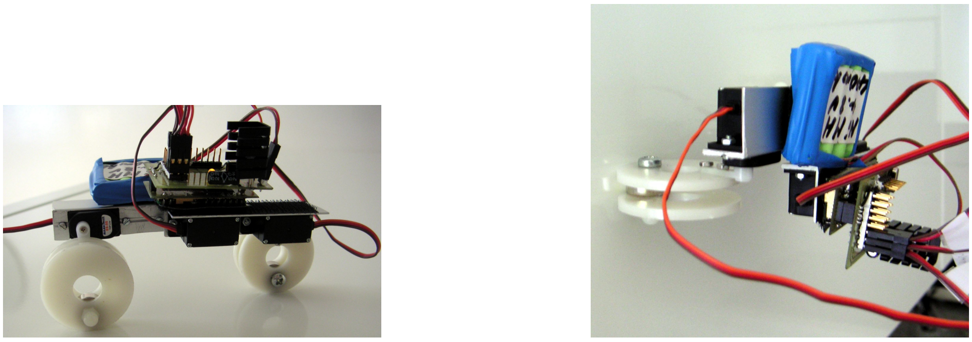


Figure 4.4: *walky* prototype

Parting in a completely different direction, a robot based on walking instead of coasting has been developed as well. The basic idea is to overcome obstacles by taking a leap. Basically there is a problem in removing a magnet from the ferromagnetic structure. Applying a perpendicular force to the magnet requires a considerable effort to get it finally in a shock-like movement off. If the magnet could be slid over an edge, or be levered off with a circular movement, the effort becomes negligible compared to the perpendicular force exerted by the magnet.

4.2.1 Mechanics

The *magnetic feet* and the *backbone* were assembled together to form the walking structure. The mechanical composition of the elements do not allow any turning of the vehicle, thus limiting its mobility. However, this prototype could be seen as half of the final robot and is only used for evaluation purposes. Refer to Section 3.1 on page 10 for details about the modules.

4.2.2 Electronics and Software

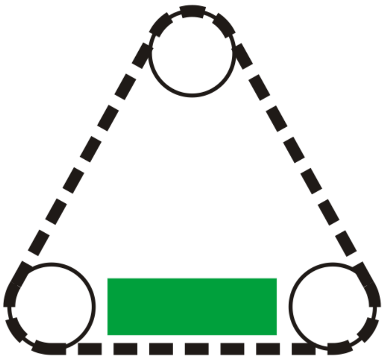
Three standard servos have been used to actuate the three joints. The C++ applet (refer to Appendix B.2 on page 37) allows to control the position by interacting with the corresponding slider. Optionally a joystick can be used (preferably one with sliders) to pilot the robot. Movement sequences have been pre-programmed allowing to take entire steps at the push of a button. Refer to Section 3.2 on page 11 for details about the controller.

4.2.3 Testing and Results

Playing with back and forth movements of the feet and bending of the back, the structure is able to slide forward or backward. It is even capable of doing so on an upside down surface. However, vertical surfaces cause problems, since the biped structure is not capable to compensate emerging momentum. If the structure were to be completed with a third foot, probably it would work just as well on vertical surfaces. Comparing the *walky* with the *caterpillar* a considerable improvement in performance can be observed. Since the walking structure is extremely simple, it seems very promising for further development.

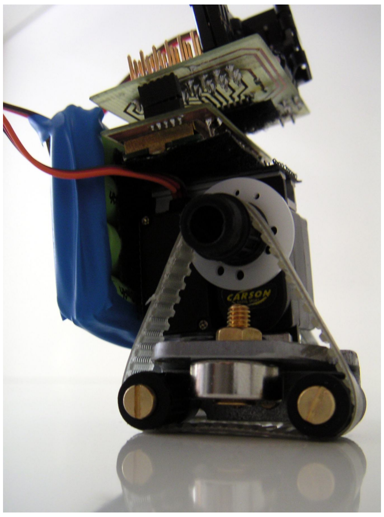
The ability to conquer corners could not be tested extensively, because the structure itself is somewhat incomplete. As described before, momentum is not well compensated by the round shape of the feet. Because of its simplicity it is also not possible to take turns. By combining two identical structures, resulting in a four legged robot, would improve both these deficiencies, and should be looked at more closely.

4.3 Tank



normal surface:	OK
vertical surface:	OK
upside down surface:	OK
corners:	gets stuck
speed:	21.74 $^{mm}/_{sec}$
mobility:	turn, rotate on spot
weight:	86 g
payload:	580 g

Figure 4.5: tank



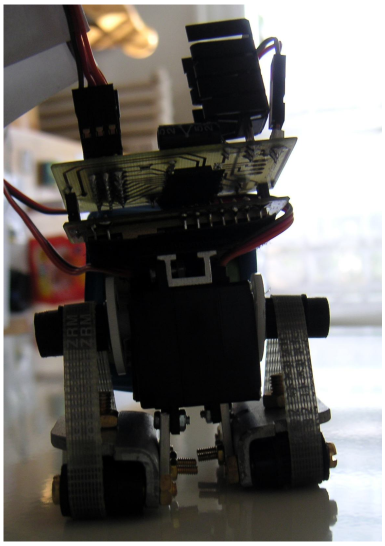


Figure 4.6: *tank* prototype

It seems there are two major directions to solve the present problem. One is to find a highly flexible and bendable mechanism, which is capable of reaching around all sorts of corners, maybe even tolerating the partial detachment of its supporting magnetic parts. This approach, as it has been attempted with the *caterpillar* prototype, shows a natural tendency to result in rather complex mechanics dotted with springs, double-joints, and so on.
On the other hand one might imagine to push miniaturisation to its limits trying to obtain a very small and extremely light robot. For such a vehicle, the obstacles in question become almost infinitely big, thus posing as simple walls and inclined planes. Even the very edges of corners become somewhat smooth transitions.

4.3.1 Mechanics

Two of the *fixed caterpillars* were assembled together to form a tank like structure. This prototype now allows to take turns, if the two modules are controlled differentially. The robot is extremely compact, bringing the center of gravity very close to the surface. Refer to Section 3.1 on page 10 for details about the modules.

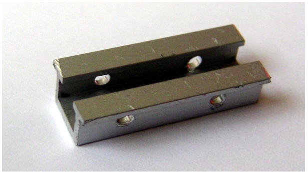


Figure 4.7: part used to link two *fixed caterpillars* together

4.3.2 Electronics and Software

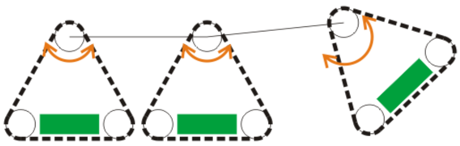
Two servos modified for continuous turning (refer to Appendix A.1 on page 29) have been used to power the caterpillar modules. The C++ applet (refer to Appendix B.2 on page 37) allows to control the speed and direction by interacting with the corresponding slider. Optionally a joystick can be used to pilot the robot. Refer to Section 3.2 on page 11 for details about the controller.

4.3.3 Testing and Results

The *tank* is now able to take turns, even to rotate on the spot by actuating its caterpillars in opposite directions. It is capable of dealing with horizontal, upside down, and vertical surfaces, which already exceeds the performances seen in *walky*. The structure is extremely compact, giving the advantage that no considerable momentums are emerging while the robot moves, especially on vertical walls. From the point of view of mobility, this structure does not need any further development.

However, some problems with dealing with corners do remain. On a convex corner the robot gets stuck, since the edge deforms the caterpillar in such a way that it gets squeezed against the magnet, thus cannot be advanced by the motor anymore. This problem could easily be solved by introducing an appropriate guidance between the two wheels. Interestingly, the robot is capable of maintaining its weight easily while "hanging" on an edge, although theoretically such a situation is extremely adverse since the magnetic flux saturates the corner of the ferromagnetic structure. Concave corners pose a problem of different nature. To overcome such an obstacle, the robot needs to move its permanent magnet away from the first surface while the front part (the supporting wheels) move along the second surface. The robot already shows tendency to do so. By adapting the caterpillar's profile, it might be possible to build up enough grip on the second surface to transmit the necessary force.

4.4 Platoon



normal surface:	OK
vertical surface:	OK
upside down surface:	OK
corners:	gets stuck
speed:	21.74 $^{mm}/_{sec}$
mobility:	straight line
weight:	138 g
payload:	620 g

Figure 4.8: platoon

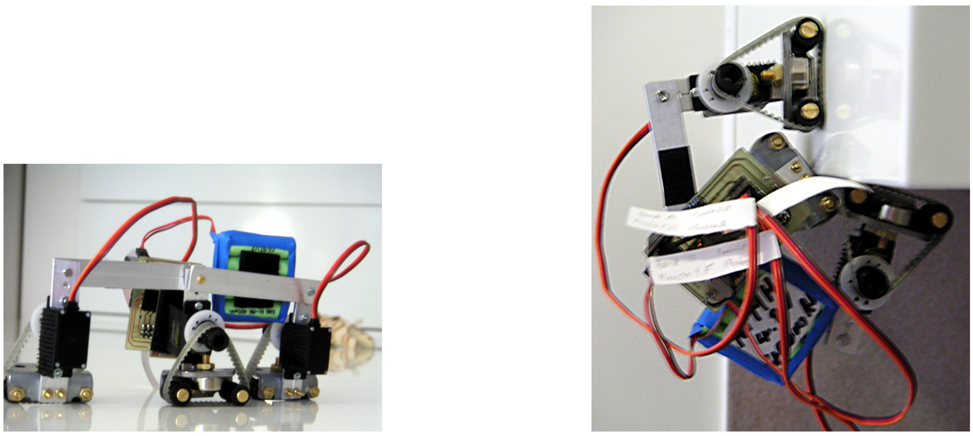


Figure 4.9: *platoon* prototype

Instead of combining the *fixed caterpillar* modules in parallel as it was the case with the *tank* prototype, they might be assembled serially as well. The individual modules help each other to pull or push the others over obstacles. Collectively, maybe they are able to push one after the other around a corner, which was not possible to achieve with the *tank*.

4.4.1 Mechanics

Three *fixed caterpillars* were assembled together standing in one line one behind the other. The joints may rotate freely in the plane comprising all the modules. The assembly itself does not allow to take turns, but again, goal is only to evaluate performance, not to build a complete robot. Refer to Section 3.1 on page 10 for details about the modules.

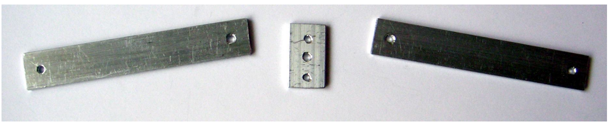


Figure 4.10: links used to assemble three *fixed caterpillars* in a line

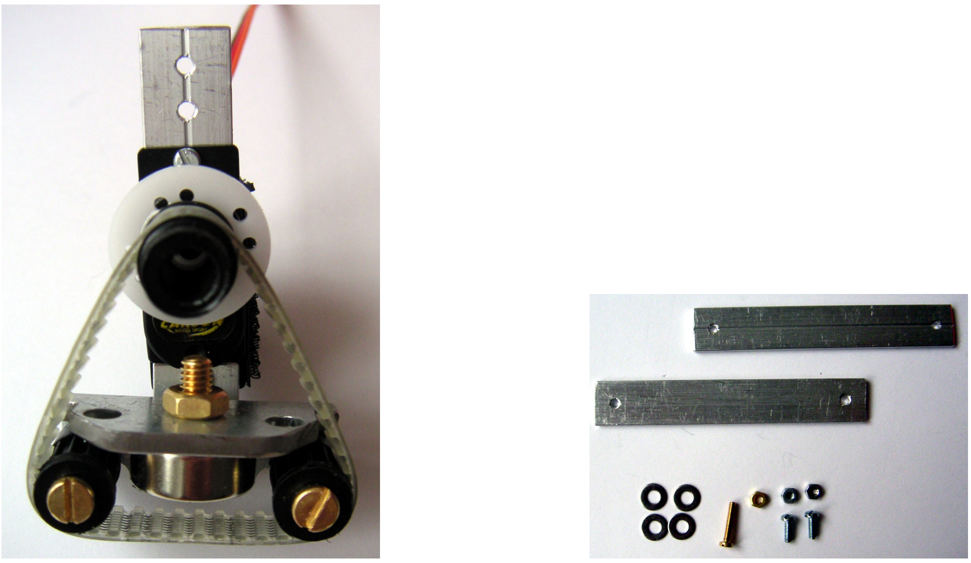


Figure 4.11: details in how to assemble the *platoon*

4.4.2 Electronics and Software

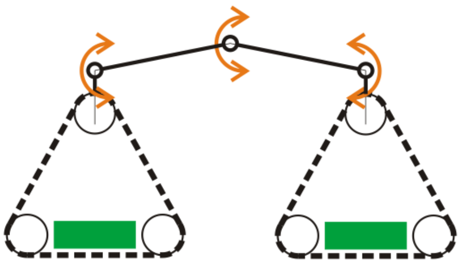
Three servos modified for continuous turning (refer to Appendix A.1 on page 29) have been used to power the caterpillar modules. The C++ applet (refer to Appendix B.2 on page 37) allows to control the speed and direction by interacting with the corresponding slider. Optionally a joystick can be used to pilot the robot. Refer to Section 3.2 on page 11 for details about the controller.

4.4.3 Testing and Results

Since the three caterpillar modules are linked together in series, only straight lines are possible. If the elements were to be replaced by the *tank* structures, and the joints between them possess a certain flexibility, it would be easy to take turns, however, rotating on the spot would not be possible, as the linking of the modules prevents it. Since the center of gravity is held extremely close to the surface, this structure is again capable of moving on horizontal, upside down, and vertical surfaces.

This structure seems to have the potential of overcoming concave corners, since the trailing elements exert enough force on the leading element to be pushed onto the new surface, a missing property of the *tank* by itself. Convex corners pose the same problem as seen with the *tank*, the caterpillar gets stuck, once it is on the verge of the corner.

4.5 Fusion One



normal surface:	OK
vertical surface:	only up/down
upside down surface:	OK
corners:	very high potential
speed:	23.81 $^{mm}/_{sec}$
mobility:	straight line
weight:	121 g
payload:	210 g

Figure 4.12: fusion one

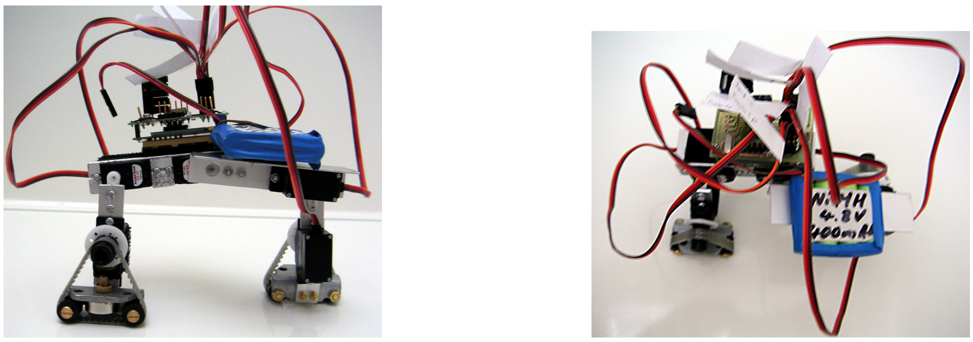


Figure 4.13: *fusion one* prototype

Trying to combine the best performances of the *fixed caterpillar* module and the jointed structure used in *walky*, a first attempt has been made with *fusion one*. On plane surfaces, the caterpillars are used to advance quickly, while the joints can be activated to help the modules to overcome edges. There are of course several possibilities to combine caterpillars with the *backbone*. One more prototype has been studied, which will be presented in the next Section.

4.5.1 Mechanics

Two of the *fixed caterpillar* modules were mounted on the *backbone* to form a structure capable of coasting while at the same time being able to actively adjust itself to the surroundings. Once more, mechanics does not allow for turning. Refer to Section 3.1 on page 10 for details about the modules.

Figure 4.14: details in how to assemble the *fusion one*

4.5.2 Electronics and Software

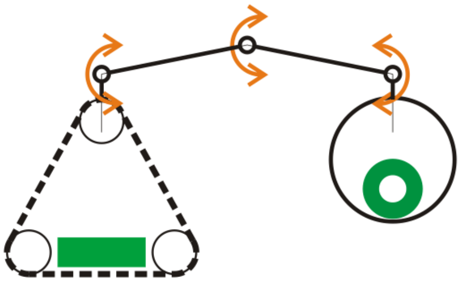
Two servos modified for continuous turning (refer to Appendix A.1 on page 29) have been used to power the robot. An additional three standard servos have been used to actuate the three joints. The C++ applet (refer to Appendix B.2 on page 37) allows to control the speed and direction by interacting with the corresponding slider. Optionally a joystick can be used (preferably one with sliders) to pilot the robot. Refer to Section 3.2 on page 11 for details about the controller.

4.5.3 Testing and Results

The structure itself is again not designed to take turns. As seen with *walky* the combination of two identical structures can easily compensate for this deficiency. *fusion one* is capable of moving on horizontal and upside down surfaces. Vertical surfaces exhibit a certain instability if the robot is moving transversally. Climbing up or down a vertical surface does not cause any problems.

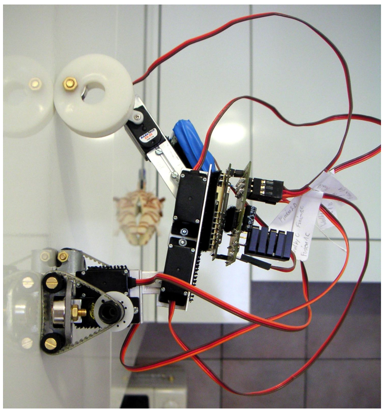
Thanks to the actively controlled joints, this prototype is capable of overcoming the convex corner, if it is carefully piloted. The success rate remains somewhat unpredictable, but by keeping in mind that all the presented robots are first generation prototypes, a very high potential of development can be seen in this solution. Concave corners are tackled a lot easier than with the preceding prototypes, since the joints can be used to help transfer the permanent magnet onto the new surface.

4.6 Fusion Two



normal surface:	OK
vertical surface:	OK
upside down surface:	OK
corners:	very high potential
speed:	23.81 $^{mm}/_{sec}$
mobility:	turn, rotate on spot
weight:	140 g
payload:	670 g

Figure 4.15: fusion two



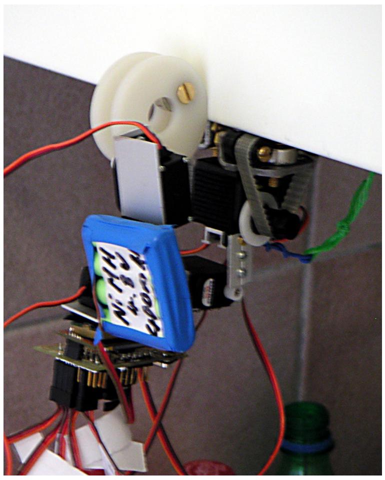


Figure 4.16: *fusion two* prototype

Trying to make use of the very simple, but quite efficient *magnetic foot* and winning back mobility with the *tank* structure, a second attempt of combining the so far studied mechanisms has been made with *fusion two*. It can mainly be seen as an extended version of the *tank*. An arm with a *magnetic foot* (which now might as well be called *magnetic hand*) is mounted on top of the *tank*, which by itself already shows extremely promising performances. The hand should be used to help the robot overcome edges by reaching out to the new surface and pulling the robot around the corner.

4.6.1 Mechanics

A *tank* subassembly composed of two *fixed caterpillar* modules was mounted at one end of the *backbone* while the other end was equipped with a *magnetic foot*. The *tank* unit is able to take turns and to rotate on the spot. The mounted arm with the *magnetic foot* at its extremity allows to grab onto surfaces to supply extra support, helping the robot to overcome corners. Refer to Section 3.1 on page 10 for details about the modules.

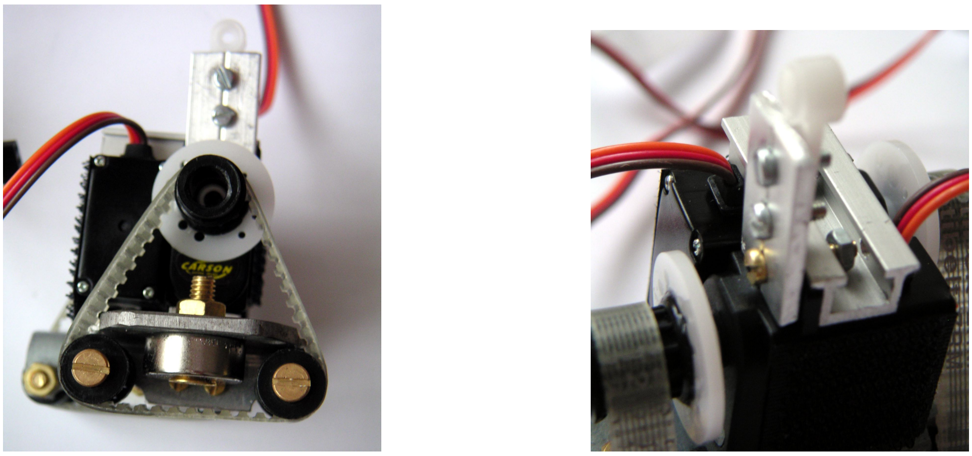


Figure 4.17: details in how to assemble *fusion one*

4.6.2 Electronics and Software

Two servos modified for continuous turning (refer to Appendix A.1 on page 29) have been used to power the robot. An additional three standard servos have been used to actuate the three joints. The C++ applet (refer to Appendix B.2 on page 37) allows to control the speed and direction by interacting with the corresponding slider. Optionally a joystick can be used (preferably one with sliders) to pilot the robot. Refer to Section 3.2 on page 11 for details about the controller.

4.6.3 Testing and Results

fusion two is again capable of taking turns and rotating on the spot. For doing so, its arm is simply lifted off the ground and only the two caterpillar elements are used. Compared to the *fusion one*, transversal movement on vertical surfaces do not pose any problems. Furthermore, the ability of moving on horizontal and upside down surfaces remains.

The arm serves as useful tool in conquering corners, since it can reach around them to give an anchor point on the new surface. The arm maneuvers need to be well coordinated with advancing, but it seems after some training, it should be possible to safely guide the robot over obstacles. Joint lengths certainly need some adjustment to render the geometry more favorable of reaching around convex corners. The possibility of deactivating single servos (for details refer to Appendix B.2 on page 37) gives the arm a certain flexibility while still aiding the vehicle to maintain its position.

Chapter 5

Ideas

In the course of this project virtually thousands of ideas and variations were found. Unfortunately time constraints did not allow the development of some of the rather promising ideas. For sake of completion they are nevertheless briefly presented in the present Chapter.

5.1 Omni-Directional Pong-Bot

Based on the already discussed *magnetic feet* (refer to 3.1 on page 10), the *omni-directional pong-bot* emerged. Thought the feet showed remarkable performances, they did have one major drawback, they were unidirectional. If a foot were to be placed at an odd angle on the surface, making contact with the circumference of one of the polymer disks first, the magnetic force is quite small. Moreover, there is no supporting feature making contact with the surface which would facilitate the orientation of the foot to a more favorable position. To compensate for this deficiency, ping-pong balls could be used instead of disks. Half a sphere were to be making contact with the surface, while a permanent magnet in the form of a ring was glued on the inside of the semi-sphere. This way, the foot is no longer planar symmetric, but axial symmetric, giving it more freedom to orient in the most favorable way from the magnetic flux point of view.

One may argue that such a foot is even worse than the already invented disk-like ones if it comes to compensating momentum. This assessment is very true, however, such feet would not be used with the *backbone* (refer to 3.1 on page 10) as it has been seen in *walky* (refer to 4.2 on page 16), but a similar structure with three arms arranged at 120°. This robot would have an omni-directional nature. Finding walking gaits for this triped would be somewhat more complex, however mobility would be increased as well, since it should be capable of changing directions and even turning on the spot, features missing in *walky*.

5.2 Spinal Cord

A *spinal cord*, as it is found in humans, presents unique features if it comes to bending and twisting. A similar structure could be imagined to solve the present problem. Each vertebra would in fact be a permanent magnet (in a ring-like shape as they can be found at [9]) equipped with two wheels, probably powered by a single motor. They are then assembled together by inserting springs between each vertebra and letting two pairs of wires, oriented at 90° to each other, run along the entire cord. By acting on one pair of wires (by means of a servo), the structure can be bent in a plane, acting on the other pair, the bending occurs in a plane perpendicular to the first one. Thus with two servos, it is possible to bend in any direction.

Since the entire cord is magnetic, it will easily stick to a ferromagnetic surface, if there is an obstacle to overcome, such as a corner, the *spinal cord* is capable of bending into the new direction. If the wheels were to be seen as feet, the entity would be very similar to a centipede.

Chapter 6

Conclusion

Developing a simple robot, piloted by means of a radio link, powered by a battery - one might even say, just developing an RC toy car as it is the dream of any little kid - proves to be more difficult than it first appears. That is of course if there is one additional constraint. The vehicle needs to be capable of moving on any ferromagnetic surface with any orientation in the room, basically on any iron based structure, as they may be found in water and steam piping systems of power plants, or inside oil tanks.

Getting a car-like robot to move on a planar surface of any orientation is a child's play. Problems arise once the vehicle has to move from one surface to a perpendicular second one. Risks are, that the robot gets stuck in the corner, that it is not capable of grasping on to the second surface, or that it simply loses its magnetic attraction and falls. Since such a rather simple obstacle already poses a big headache, obstructions of any shape are left aside for the moment. In fact, the final goal of the global project demands an extremely small robot, which means that almost any obstacle found in the field of application becomes a combination of intersecting surfaces at different angles usually around 90°.

Although some of the testing and developing could not have been carried out due to some unexpected shipment delays, the over all results are satisfying. Especially the *fusion two* robot (refer to 4.6) on page 24 shows a big potential of improvement to master the angle problem. It already shows an exceptional mobility and seems fairly easy to operate.

It is suggested that the development of the *fusion one* and *fusion two* prototypes is carried on, since these seem to be the most promising ideas. Possibly there is a way of using genetic algorithms and evolutionary theories to find the most performing solution in therms of combination of the modules and control sequences. However, it seems quite difficult to define an appropriate fitness function.

With the collection of modules which have been constructed, subsequent projects may find even other combinations or improvements, which have not been covered by the present work. Imagination is the limit!

Pascal Gilbert
pascal.gilbert@epfl.ch

Appendix A

Hardware

A.1 Modifying a Servo Motor for continuous turning[1]

The trick is to make the servo think that the output shaft is always at the 90° mark. This is done by removing the feedback potentiometer, and replacing it with a symmetric resistive bridge producing the signal of the 90° position. Thus, imposing the signal for 0° will cause the motor to turn on full speed in one direction. The signal for 180° will cause the motor to go the other direction. Since the feedback from the output shaft is disconnected, the servo will continue in the appropriate direction as long as the signal remains.

As for the details, there are actually only two modifications to make to the servo

1. Replace the position sensing potentiometer with an equivalent resistor network
2. Remove the mechanical stop from the output shaft

To get started, follow these steps

- Open the case by removing the 4 screws located at the bottom of the servo. The bottom plate should come off easily. Remove the top of the case. You will find a set of gears under the top case, a several blobs of white grease. Try hard to save the grease by leaving it on the gears.
- Be careful to note how the gears are arranged, and remove them from the top of the servo.
- Locate and remove any small philips head screws which hold the electronics assembly in place. Several variants of servos exist, some might not have any additional screws but are clipped together.
- Remove the circuit board from the case.
- Desolder the potentiometer from the board.

[1] based on `http://www.seattlerobotics.org/guide/servohack.html`

- Wire in the resistor network. To do this, place the resistors side by side and twist one pair of leads. Solder them together, but leave one of the leads long enough to make a 3 wire part. Then replace the potentiometer with this 3 wire network.
- Reassemble the circuit board into the case.
- Locate the mechanical stops, they are either directly on the output shaft, or inside the top housing. Use a sharp knife to cut down flush the tab of plastic.
- Replace the gears as they were, replace the top of the case, the bottom plate, and screw everything back together.

Figure A.1: disassembled servo motor

Appendix B

Software

B.1 Servo Control

The following flowchart demonstrates how it is possible to control several servo lines using only two timers

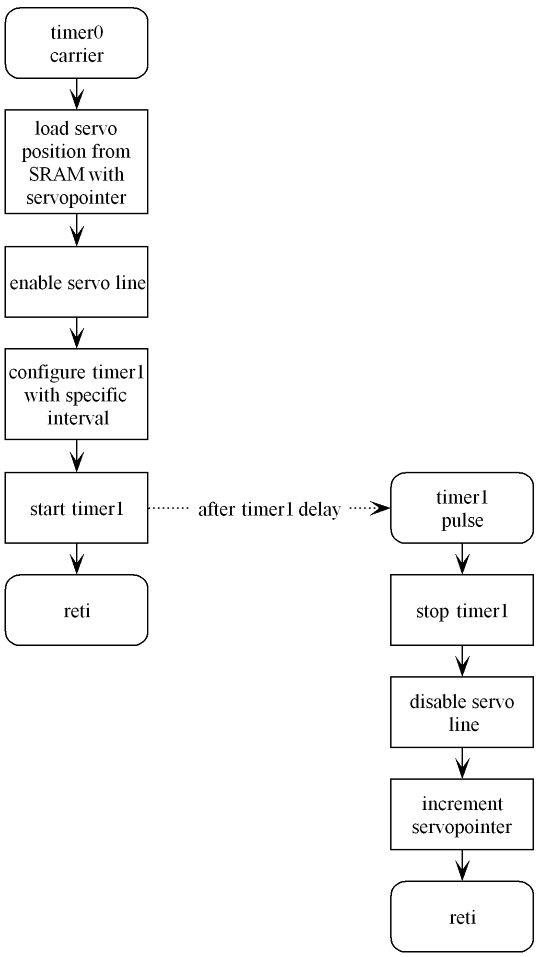


Figure B.1: servo control mechanism flowchart

B.1.1 MagBotBrain.asm

```
; MagBotBrain
; Version 1.0
; (c) 2007 by Freddy


; === basic includes ===

.include        "tiny2313def.inc"         ; include AVR port/bit definitions
.include        "servo_definitions.asm"   ; include register/constant definitions
.include        "servo_macros.asm"        ; include macro definitions
.include        "servo_sram.asm"          ; include SRAM initialisation


; === interrupt vector table ===

.org    0                                  ; reset vector
        rjmp    reset
.org    OVF1addr                           ; Overflow 1 Interrupt Vector
        rjmp    pulse
.org    OVF0addr                           ; Overflow 0 Interrupt Vector
        rjmp    carrier
.org    URXCaddr                           ; UART receive complete Interrupt Vector
        rjmp    UART_rxc

.org    0x13                               ; end of vector table


; === interrupt service routine ===

.include        "servo_interrupts.asm"    ; include Interrupt Service Routines


; === initialisation ===

reset:  OUTI    SPL,RAMEND                 ; Load Stack Pointer (SP)

        OUTI    DDRB,0xff                  ; configure PortB: servos output
        OUTI    DDRD,0x40                  ; configure PortD: LED output, rest input

        clr     zero                       ; prepare zero register

; *** servo start value initialisation ***
;       ldi     w,0x7f                     ; set all to 127 (middle position)
;       sts     servo1,w
;       sts     servo2,w
;       sts     servo3,w
;       sts     servo4,w
;       sts     servo5,w
;       sts     servo6,w
;       sts     servo7,w
;       sts     servo8,w
        sts     s_ctrl,zero                ; disable all servos

; *** initialize timers ***

        ldi     w,0x03                     ; CK/64 @16MHz --> 510 => 2.04 msec pulse
        mov     scaler,w                   ; load prescaler preset for timer1

        out     TCCR1A,zero                ; normal port operation
        out     TCCR1B,zero                ; CS1=0 STOP (prescaler timer1)
        out     TCCR0A,zero                ; normal port operation
        OUTI    TCCR0B,4                   ; CS0=4 CK/256 (prescaler timer0)

        OUTI    TIMSK,(1<<TOIE1)+(1<<TOIE0)
                                           ; timer1 and timer0 overflow interrupt enable
        ldi     w,(-156)                   ; timing for carrier wave (2.496 msec * 8 = 19.968) @ 16MHz Clock
        mov     timer,w
        out     TCNT0,timer                ; preset timer0 for definit intervall

        clr     s_p                        ; clear servo pointer

; *** UART Initialisation ***
        OUTI    UBRRH,0x00
;       OUTI    UBRRL,0x19                 ; baud rate 38400
        OUTI    UBRRL,0x33                 ; baud rate 19200
        ldi     w,(1<<RXCIE)+(0<<TXCIE)+(0<<UDRIE)+(1<<RXEN)+(1<<TXEN)+(0<<U2X)
        out     UCSRB,w                    ; rx interrupt enabled, rx and tx enabled
        ldi     w,(1<<UCSZ1)+(1<<UCSZ0)
        out     UCSRC,w                    ; no stop bit, no parity, 8-bit



; *** final boot sequence ***
        clr     sys                        ; clear system flags
        sei                                ; set global interrupt enable
        sbi     PORTD,6                    ; clear LED
        WAIT_MS 50
        cbi     PORTD,6                    ; set LED
        WAIT_MS 50
        sbi     PORTD,6                    ; clear LED
        WAIT_MS 50
        cbi     PORTD,6                    ; set LED
        WAIT_MS 50
        sbi     PORTD,6                    ; clear LED
```

```
        rjmp    main                    ; commence operation


; === main code ===

main:   INVP    PORTD,6                 ; toggle LED
        WAIT_MS 100                     ; stay alive

        rjmp    main                    ; loop back


; === lookup tables ===

smask:                                  ; servopointer to servoline translation masks
.db     0b00000001,0b00000010,0b00000100,0b00001000,0b00010000,0b00100000,0b01000000,0b10000000
```

B.1.2 servo_interrupts.asm

```
; file: servo_interrupts.asm
; MagBotBrain tiny2313 interrupt service routines (ISR)
; copyright (c) 2007 Freddy
; 2007-05-25


; *** servo position interpreter ***
pulse:
        in      _sreg,SREG              ; save context

;       sei                             ; enable nested interrupts

        out     TCCR1B,zero             ; CS0=0 STOP (prescaler timer0)
        out     PORTB,zero              ; disable sensorline
        INC_CYC s_p,0,7                 ; cyclic servo update

        out     SREG,_sreg              ; restore context
        reti

; *** carrier wave generator ***
carrier:
        in      _sreg,SREG              ; save context
        push    char
;       PUSH4   a0,a1,a2,a3             ; ATTENTION nested interrupts
        PUSHZ                           ; save context

;       sei                             ; enable nested interrupts

        out     TCNT0,timer             ; preset timer0 for defined intervall

        ldi     a0,trim                 ; offset (global servo-trim = 1msec)
        clr     a1                      ; clear high byte

        ldi     zl, low(servo1)
        ldi     zh,high(servo1)         ; load SRAM entrypoint
        add     zl,s_p
        sbci    zh,-0x00                ; increment to corresp. servo value
        ld      a2,Z                    ; load
        clr     a3                      ; clear high byte

        add     a0,a2                   ; add low byte
        adc     a1,a3                   ; add high byte with carry --> timer-value

        lsl     a0
        rol     a1                      ; multiply 16-bit value by two

        neg     a0
        com     a1                      ; negate 16-bit value

        nop
        nop                             ; some delay for whatever reason !!!???

        ldi     zl, low(2*smask)
        ldi     zh,high(2*smask)        ; load table entrypoint
        add     zl,s_p
        sbci    zh,-0x00                ; increment to corresp. mask in lookup table
        lpm                             ; load mask into char (r0)

        lds     a2,s_ctrl               ; get servo control bits (START/STOP)
        and     char,a2                 ; apply control to servoline

        out     PORTB,char              ; enable servoline

        out     TCNT1H,a1               ; set high byte servo timing on timer1
        out     TCNT1L,a0               ; set low byte servo timing on timer1
        out     TCCR1B,scaler           ; prescaler timer1 (activate timer)


        POPZ
;       POP4    a0,a1,a2,a3
        pop     char
        out     SREG,_sreg              ; restore context
        reti

; *** UART receive complete Interrupt Service Routine ***
UART_rxc:
```

```
        in      _sreg ,SREG
;       push    _w
;       push    _u
;       PUSHY                           ; save context

;       INVP    PORTD, 6                ; toggle LED

        in      _w ,UDR                 ; read received data
        sbrc    sys , SecByte           ; Second Byte?
        rjmp    U_1                     ; jump ahead to store second byte
        sbrc    sys , TriByte           ; Third Byte?
        rjmp    U_2                     ; jump ahead to store third byte
        sbrc    sys , QuadByte          ; Fourth Byte?
        rjmp    U_3                     ; jump ahead to store fourth byte

U_0:    cpi     _w ,0x09                ; compare with 9
        brge    U_done                  ; leave if invalid memory address
        sts     rx_red , _w             ; store first byte
        ori     sys ,(1<<SecByte)       ; set Second Byte Flag
        rjmp    U_done                  ; leave interrupt routine

U_1:    andi    sys ,~(1<<SecByte)      ; clear Second Byte Flag
        lds     _u , rx_red             ; get redundancy value
        cp      _u , _w                 ; compare two received bytes
        brne    U_done                  ; branch if not equal , −−> error , leave interrupt
        mov     v_p , _w                ; store Memory Offset
        ori     sys ,(1<<TriByte)       ; set Third Byte Flag
        rjmp    U_done                  ; leave interrupt routine

U_2:    andi    sys ,~(1<<TriByte)      ; clear Third Byte Flag
        sts     rx_red , _w             ; store Third byte
        ori     sys ,(1<<QuadByte)      ; set Fourth Byte Flag
        rjmp    U_done                  ; leave interrupt routine

U_3:    andi    sys ,~(1<<QuadByte)     ; clear Fourth Byte Flag
        lds     _u , rx_red             ; get redundancy value
        cp      _u , _w                 ; compare two received bytes
        brne    U_done                  ; branch if not equal , −−> error , leave interrupt
        ldi     yl , low(servo1)
        ldi     yh , high(servo1)       ; load SRAM entrypoint
        add     yl , v_p
        sbci    yh, −0x00               ; increment to corresponding memory address
        st      Y, _w                   ; store value

U_done: ;POPY
;       pop     _u
;       pop     _w
        out     SREG, _sreg             ; restore context
        reti                            ; leave interrupt routine
```

B.1.3 servo_sram.asm

```
; file : servo_sram.asm
; MagBotBrain tiny2313 SRAM initialisation
; copyright (c) 2007 Freddy
; 2007−05−25

; *** SRAM definition ***

.dseg                                   ; 19 bytes used of 128 bytes total
servo1: .byte   1                       ; Servo 1 position                      + 0
servo2: .byte   1                       ; Servo 2 position                      + 1
servo3: .byte   1                       ; Servo 3 position                      + 2
servo4: .byte   1                       ; Servo 4 position                      + 3
servo5: .byte   1                       ; Servo 5 position                      + 4
servo6: .byte   1                       ; Servo 6 position NOT IN USE           + 5
servo7: .byte   1                       ; Servo 7 position NOT IN USE           + 6
servo8: .byte   1                       ; Servo 8 position NOT IMPLEMENTED YET  + 7

s_ctrl: .byte   1                       ; Servo Control                         + 8

rx_red: .byte   1                       ; redundancy byte (UART reception)
.cseg
```

B.1.4 servo_definitions.asm

```
; file : servo_definitions.asm
; MagBotBrain tiny2313 definitions
; copyright (c) 2007 Freddy
; 2007−05−25


; === definitions ===
.nolist                 ; do not include in listing
.set    clock   = 16000000

.def    char    = r0    ; character (ASCII)
.def    _sreg   = r1    ; saves the status during interrupts
.def    _u      = r2    ; saves working reg u during interrupt
.def    u       = r3    ; scratch register (macros, routines)

.def    zero    = r4    ; used as zero register (read only!!)
.def    scaler  = r5    ; timer1 prescaler preset
```

```
.def    timer   = r6    ; well-defined timer0 intervall storage

.def    v_p     = r7    ; UART value pointer
;.def   not used= r8
;.def   not used= r9
;.def   not used= r10
;.def   not used= r11
;.def   not used= r12
;.def   not used= r13
;.def   not used= r14
;.def   not used= r15

.def    w       = r16   ; working register for macros
.def    _w      = r17   ; working register for interrupts

.equ    a       = 18
.def    a0      = r18   ; 4-byte register a
.def    a1      = r19
.def    a2      = r20
.def    a3      = r21

.equ    b       = 22
.def    b0      = r22   ; 4-byte register b
.def    b1      = r23
.def    b2      = r24
.def    b3      = r25

.def    s_p     = r26   ; Servo Pointer (for servo service timer)
.def    sys     = r27   ; System Flag register

.equ    y       = 28    ; pointer y
.equ    z       = 30    ; pointer z


; === ServoController definitions ===

; *** System Flags ***
.equ    SecByte = 0     ; receive second byte
.equ    TriByte = 1     ; receive third byte
.equ    QuadByte= 2     ; receive fourth byte
;.equ   reserved= 3     ; not in use
;.equ   reserved= 4     ; not in use
;.equ   reserved= 5     ; not in use
;.equ   reserved= 6     ; not in use
;.equ   reserved= 7     ; not in use

; *** Servo Definitions ***
.set    trim    =57     ; global servo trim (=0.228msec)


; === ASCII codes
.equ    BEL     =0x07   ; bell
.equ    HT      =0x09   ; horizontal tab
.equ    TAB     =0x09   ; tab
.equ    LF      =0x0a   ; line feed
.equ    VT      =0x0b   ; vertical tab
.equ    FF      =0x0c   ; form feed
.equ    CR      =0x0d   ; carriage return
.equ    SPACE   =0x20   ; space code
.equ    DEL     =0x7f   ; delete
.equ    BS      =0x08   ; back space
.list
```

B.1.5 servo_macros.asm

```
; file: servo_macros.asm
; MagBotBrain tiny2313 MACROS
; copyright (c) 2007 Freddy
; 2007-05-25


.macro  PUSHZ                   ; push Z
        push    zl
        push    zh
.endmacro

.macro  POPZ                    ; pop Z
        pop     zh
        pop     zl
.endmacro

.macro  PUSH4
        push    @0
        push    @1
        push    @2
        push    @3
.endmacro
.macro  POP4
        pop     @3
        pop     @2
        pop     @1
        pop     @0
.endmacro

.macro  OUTI    ; port,k        output immediate value to port
```

```
        ldi     w,@1
        out     @0,w
.endmacro

.macro  LOOKUP2 ;r1,r0, index,tbl
        mov     zl,@2               ; move index into z
        clr     zh
        lsl     zl                  ; multiply by 2
        rol     zh
        subi    zl, low(-2*@3)      ; add base address of table
        sbci    zh,high(-2*@3)
        lpm                         ; get LSB byte
        mov     w,r0                ; temporary store LSB in w
        adiw    zl,1                ; increment Z
        lpm                         ; get MSB byte
        mov     @0,r0               ; mov MSB to res1
        mov     @1,w                ; mov LSB to res0
.endmacro

.macro  INC2
        ldi     w,0xff
        sub     @1,w
        sbc     @0,w
.endmacro

.macro  INC_CYC ; reg,low,high
        cpi     @0,@2
        brsh    _low                ; reg>=high then reg=low
        cpi     @0,@1
        brlo    _low                ; reg< low  then reg=low
        inc     @0
        rjmp    _done
_low:   ldi     @0,@1
_done:
.endmacro

.macro  INVP    ; port,bit          ; inverse port,bit
        sbis    @0,@1
        rjmp    PC+3
        cbi     @0,@1
        rjmp    PC+2
        sbi     @0,@1
.endmacro

; wait micro-seconds (us)
; us = x*3*1000'000/clock)______==>_x=us*clock/3000'000
.macro  WAIT_US ; k
        ldi     w, low((clock/1000*@0/3000)-1)
        mov     u,w
        ldi     w,high((clock/1000*@0/3000)-1)+1 ; set up: 3 cyles
        dec     u
        brne    PC-1                ; inner loop: 3 cycles
        dec     u                   ; adjustment for outer loop
        dec     w
        brne    PC-4
        .endmacro

; wait mili-seconds (ms)
.macro  WAIT_MS ; k
        ldi     w, low(@0)
        mov     u,w                 ; u = LSB
        ldi     w,high(@0)+1        ; w = MSB
wait_ms:
        push    w                   ; wait 1000 usec
        push    u
        ldi     w, low((clock/3000)-5)
        mov     u,w
        ldi     w,high((clock/3000)-5)+1
        dec     u
        brne    PC-1                ; inner loop: 3 cycles
        dec     u                   ; adjustment for outer loop
        dec     w
        brne    PC-4
        pop     u
        pop     w

        dec     u
        brne    wait_ms
        dec     w
        brne    wait_ms
.endmacro
```

B.2 Visual C++ Applet[7][3]

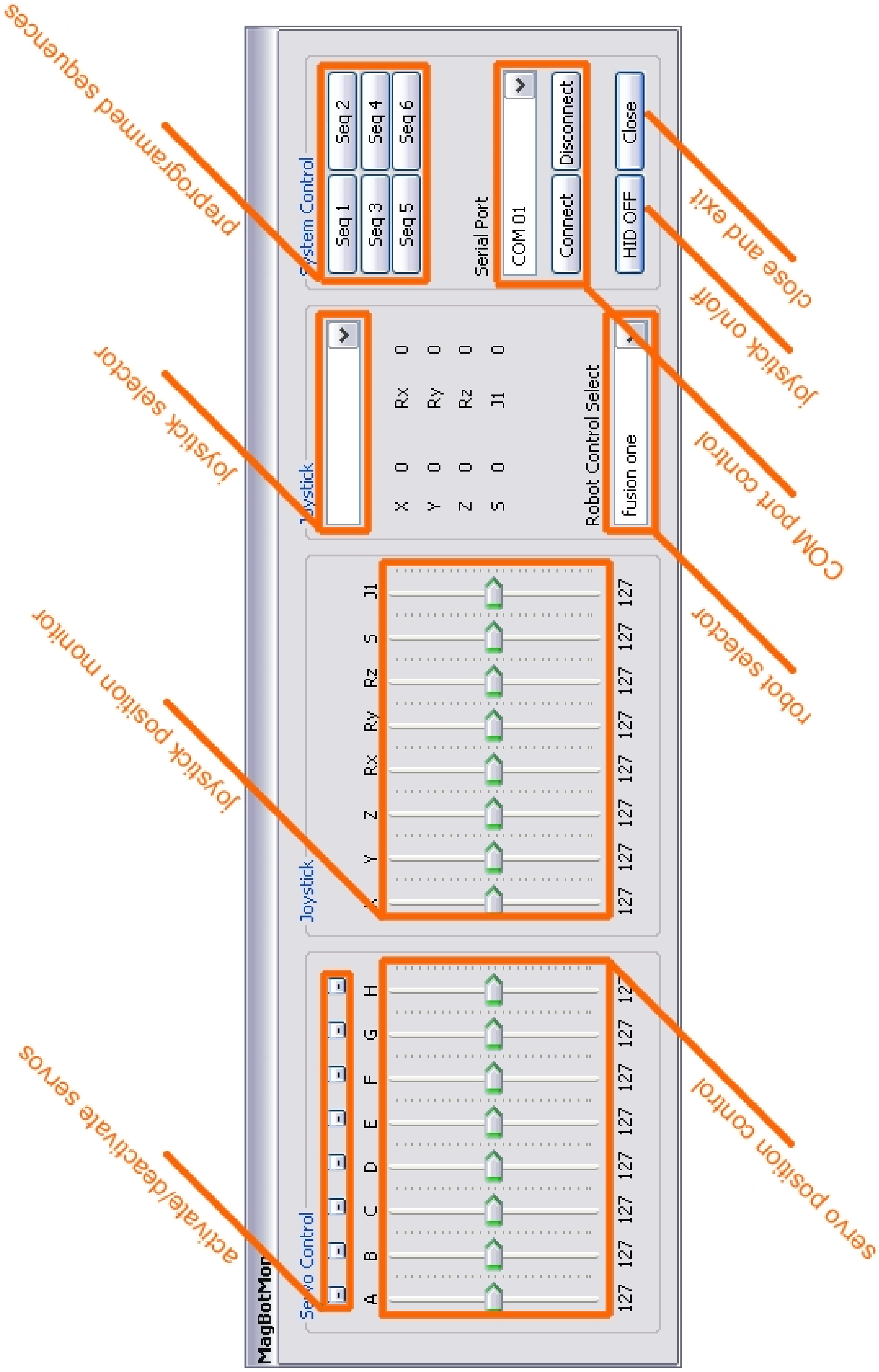


Figure B.2: MagBotMon interface

Appendix C

Electronics

C.1 Schematics

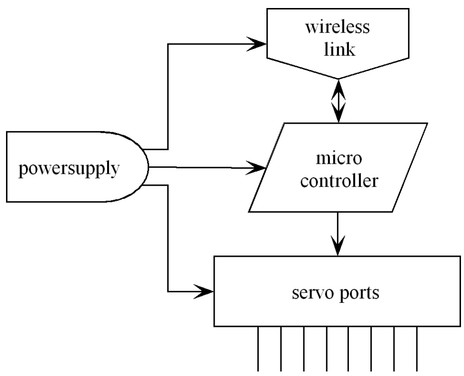


Figure C.1: component block diagram

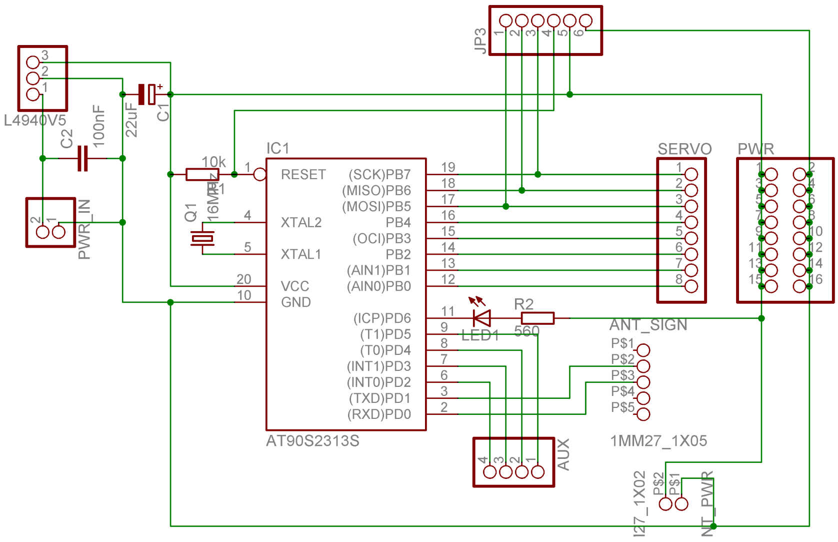


Figure C.2: schematics MagBoard

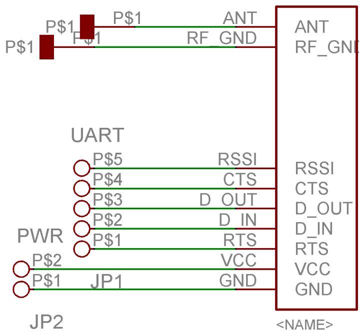
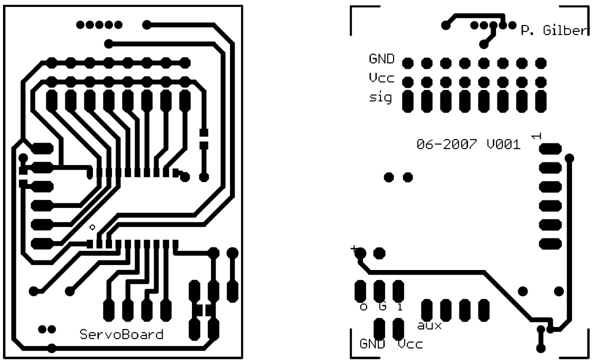


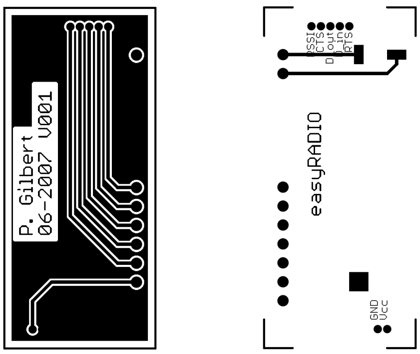
Figure C.3: schematics easyRADIO

C.2 Prints



(a) top (b) bottom

Figure C.4: MagBoard



(a) top (b) bottom

Figure C.5: easyRADIO

Bibliography

[1] ATMEL. http://www.atmel.com/.

[2] Vincent Chenal. Miniature inspection robot in ferromagnetic environments. Semester project, Hannes Bleuler, Francesco Mondada, Tarek Baaboura, January 2007.

[3] DirectInput Joystick C++ class. http://www.btinternet.com/~jason.brooks/Programming/DirectX/DirectInput/CDIJoystick.htm.

[4] Christophe Groux. Micro-robot d'inspection à roues magnétique. Semester project, Roland Sigwart, Fabien Tâache, Francesco Mondada, June 2006.

[5] Wikipedia: Lenz's Law. http://en.wikipedia.org/wiki/Lenz%27s_law.

[6] LPRS. http://www.lprs.co.uk/main/product.info.php?productid=154.

[7] serial port C++ class. http://www.codeproject.com/system/Serial.asp.

[8] Kiyoshi Tsuru Shinya Kitai and Shigeo Hirose. The proposal of swarm type wall climbing robot system "anchor climber". Technical report, Department of Mechanical and Aerospace Engineering, Tokyo Institute of Technology.

[9] SuperMagnete. http://www.supermagnete.ch.

[10] Zeliang Xu and Peisun Ma. A wall-climbing robot for labelling scale of oil tank's volume. Technical Report 4, The Robotics Institute, School of Mechanical Engineering, shanghai Jiao Tong University, 1954 Huashan Road, Shanghai, October 2001.

[11] K. Iwao Y. Kawaguchi, I. Yoshida and T. Kikuta. Sensors and crabbing for an in-pipe magnetic-wheeled robot. Technical report, Osaka Gas Co., Ltd., 6-19-9 Torishima, Konohana, Osaka, 554 JAPAN.