The Man Behind the dsNavCon Design

An Interview with Guido Ottaviani

Guido Ottaviani worked as an analog and digital developer for several years for an Italian communications company. Today, he is a technical manager for a large Italian editorial group. In his spare time, Guido designs autonomous robots, shares his experiences with other self-professed "electronics addicts," and evangelizes robot design. In June, Guido and I discussed his fascination with robotics, his favorite microcontrollers, and some of his innovative projects, such as the dsNavCon system.—Nan Price, Associate Editor

NAN: Where do you live?

GUIDO: I live in wonderful Rome, Italy, the eternal city. Very chaotic but very stimulating, according to many points of view.

NAN: How did you become interested in electronics?

GUIDO: I was 13 at junior high school. A substitute teacher went out of the standard program (technical drawing) and started explaining the Crystal Set radio to us. I was so fascinated by that new world that, four years later, I got a radio amateur license, after a period of SWL activity and the construction of some tube devices as audio amplifiers and long-wave band receivers.

I knew all my youth friends on air, all electronics addicts like me. We spent most of our time in experimenting something.

A little anecdote, just to understand who we were: one afternoon after school, there were four of us in the laboratory trying to make an HF transmitter function. It was built by installing some tubes on an upside-down aluminum baking pan used as the base. Eventually, the transmitter started outputting some power to the load, an incandescent lamp used instead of the antenna. While we were still celebrating our success, looking at the lamp glowing to the rhythm of the modulating voice, our friend's mother came into the room and asked, "Why are you so happy? All this stuff just to light up a lamp?"

That episode didn't decrease our passion, and this passion wasn't an obstacle in my regular life: my wife is the sister of one of those fellows.

NAN: What was the first MCU you worked with? Where were you at the time? Tell us about the project and what you learned.

Guido's workstation

GUIDO: The very first one was not technically an MCU, that was too early. It was in the mid 1980s. I worked on an 8085 CPU-based board with a lot of peripherals, clocked at 470 kHz (less than half a megahertz!) used for a radio set control panel. I was an analog circuits designer in a big electronics company, and I had started studying digital electronics on my own on a Bugbook series of self-instruction books, which were very expensive at that time. When the company needed an assembly programmer to work on this board, I said, "Don't worry, I know the 8085 CPU very well." Of course this was not true, but they never complained, because that job was done well and within the scheduled time.

I learned a lot about how to optimize CPU cycles on a slow processor. The program had very little time to switch off the receiver to avoid destroying it before the powerful transmitter started.

Flow charts on paper, a Motorola developing system with the program saved on an 8" floppy disk, a very primitive character-based editor, the program burned on an external EPROM and erased with a UV lamp. That was the environment! When, 20 years later, I started again with embedded programming for my hobby, using Microchip Technology's MPLAB IDE (maybe still version 6.xx) and a Microchip Technology PIC16F84, it looked like paradise to me, even if I had to relearn almost everything.

But, what I've learned about code optimization—both for speed and size—is still useful even when I program the many resources on the dsPIC33F series.

NAN: What is your current occupation?

GUIDO: After a good period of electronic designing in that electronics company, a colleague of mine and I decided to face up to a challenge. There was the need to build up the technical department for a newspaper company from scratch. Since then, I've learned a lot about editorial systems, picture editing, page layout, computer to plate, paper, ink, web presses, and every technical issue needed to produce and print newspapers and magazines. Now I'm managing a company that takes care of all the technologies for the entire editorial group.

NAN: You worked in the field of analog and digital development for several years. Tell us a bit about your background and experiences.

GUIDO: Let me talk about my first day of work, exactly 31 years ago.

Being a radio amateur and electronics fan, I went often to the surplus stores to find some useful components and devices, or just to touch the wonderful receivers or instruments: Bird wattmeters, Collins or Racal receivers, BC 312, BC 603 or BC 1000 military receivers and everything else on the shelves.

The first day of work in the laboratory they told to me, "Start learning that instrument." It was a Hewlett-Packard spectrum analyzer (maybe an HP85-something) that cost more than 10 times my annual gross salary at that time. I still remember the excitement of being able to touch it, for that day and the following days. Working in a company full of these kinds of instruments (the building even had a repair and calibration laboratory with HP employees), with more than a thousand engineers who knew everything from DC to microwaves to learn from, was like living in Eden. The salary was a secondary issue (at that time).

I worked on audio and RF circuits in the HF to UHF bands: active antennas, radiogoniometers, first tests on frequency hopping and spread spectrum, and a first sample of a Motorola 68000-based GPS as big as a microwave oven.

Each instrument had an HPIB (or GPIB or IEEE488) interface to the computer. So I started approaching this new (for me) world of programming an HP9845 computer (with a cost equivalent to 5 years of my salary then) to build up automatic test sets for the circuits I developed. I was very satisfied when I was able to obtain a 10-Hz frequency hopping by a Takeda-Riken frequency synthesizer. It was not easy with such a slow computer, BASIC language, and a bus with long latencies. I had to buffer each string of commands in an array and use some special precaching features of the HPIB interface I found in the manual.

Every circuit, even if it was analog, was interfaced with digital ports. The boards were full of SN74*xx* (or SN54*xx*) ICs, just to make some simple functions as switching, multiplexing, or similar. Here again, my lack of knowledge was filled with the "long history experience" on Bugbook series.

Well, audio, RF, programming, communications, interfacing, digital circuits. What was I still missing to have a good background for the next-coming hobby of robotics? Ah! Embedded programming. But I've already mentioned this experience.

After all these design jobs, because my knowledge started spreading on many different projects, it was natural to start working as a system engineer, taking care of all the aspects of a complex system, including project management.

NAN: You have a long-time interest in robotics and autonomous robot design. When did you first become interested in robots and why?

GUIDO: I've loved the very simple robots in the toy store windows since I was young, the same I have on my website (Pino and Nino). But they were too simple. Just making something a little bit more sophisticated required too much electronics at that time.

After a big gap in my electronics activity, I discovered a newly published robotic magazine, with an electronic parts supplement. This enabled me to build a programmable robot based on a Microchip PIC16F84. A new adventure started. I felt much younger. Suddenly, all the electronics-specialized neurons inside my brain, after being asleep for many years, woke up and started running again. Thanks to the Internet (not yet available when I left professional electronics design), I discovered a lot of new things: MCUs, free IDEs running even on a simple computer, free compilers, very cheap programming devices, and lots of documentation freely available. I threw away the last Texas Instruments databook I still had on my bookshelf and started studying again. There were a lot of new things to know, but, with a good background, it was a pleasant (if frantic) study. I've also bought some books, but they became old before I finished reading them.

Within a few months, jumping among all the hardware and software versions Microchip released at an astonishing rate, I found Johann Borenstein et al's book *Where Am I?: Systems and Methods for Mobile Robot Positioning* (University of Michigan, 1996). This report and Borenstein's website taught me a lot about autonomous navigation techniques. David P. Anderson's "My Robots" webpage (www.geology.smu.edu/~dpa-www/myrobots.html) inspired all my robots, completed or forthcoming.

I've never wanted to make a remote-controlled car, so my robots must navigate autonomously in an unknown environment, reacting to the external stimuli.

NAN: On your website, you include information and pictures from several robotics events and competitions. Do you enter competitions or are you involved in organizing the events?

GUIDO: I've participated in some of the competitions, but that is not my main goal. The first challenge is always with me. I set my target and do my work until it is achieved. The first target was the SR04 robot from Anderson. It included most of the concepts you must know to make an autonomous robot.

I first tried to build a robot for the Explorer category competitions. You can see some slide shows on my website. This was very popular among the events organized by technical

Massiomo Banzi (left) and Guido Ottaviani at the Romecup 2011

schools, both in the junior (no programming) and the senior version (programmable robots). But that didn't fit my target; it requires more speed than precision. I tried, together with some friends, to involve more people in organizing an "indoor precision competition." We also defined the name, the logo, and the rules. It was RTC, which stood for Robo-Tolomeo Cup, with the name of the big greek cartographer, to prove that this challenge could be a simpler precursor for the better known, more complicated, outdoor Robo-Magellan from Seattle Robotics. But this was not so spectacular, and we were not able to involve enough people.

I won my personal competition when I was able to complete a full UMBmark (the video is on my website) with a centimeter precision. Then, that project was over and I had a new target: the Anderson's jBot (always him!). Right now, I'm designing the foundation of an outdoor robot, Lino.

But I usually participate—at least as an observer—in every competition in Italy I am able to. It is always stimulating, and I always learn a lot over there. Furthermore, I've had the opportunity to meet again the many people I've known in the past years, often thanks to the popularity *Circuit Cellar* gave to me. We are a bunch of robotics fans, spread all over Italy, and in touch every day via the 'Net. Those are the only moments we can meet personally. Each meeting, after a lot of robotics, always ends up in a celebration around some food.

This is one of the other disciplines typically involved in robotics: friendship.

NAN: Speaking of robotics events in Italy, can you tell us about the electronics "scene" in Italy? For instance, in the U.S., hackspaces and "circuit cellars" are cropping up everywhere, from New York to small towns in the southern states to Silicon Valley. These small groups comprise enthusiastic electronics designers of all ages and backgrounds.

GUIDO: The scene is very confused right now. The interest in robotics (much more than just electronics) is growing very quickly, and a lot of different facts pop up every day. One common element is actually the schools. A lot of teachers are involving more and more students on these projects. They understand the potential of robotics. The appeal it has for young people is very useful to stimulate them in a lot of different disciplines. It's easier to explain math, physics, science, and a lot of other things, by immediately using that theory in a real, practical robot. Some university students are showing the teachers how they can teach to the younger students. Those projects are new for most of the current teaching staff, and they are putting in a lot of personal effort to be up-to-date. I've talked with many teachers, one uses his own garage to receive the students in the afternoon and build their robots for an upcoming competition. The school structures have no money to let the laboratories open after hours.

Some association started organizing. "Mondo Digitale" is a group of people who are trying to define the way schools can proceed. The "Romecup 2012" was crowded with a lot of students of all ages, most of them very enthusiastic about robots. It was amazing to listen to a very young boy who precisely explained to me everything about sensors, actuators, and central processors for his robot.

Also Massimo Banzi, participating in these events, evangelizing Arduino and creating the "Officine Arduino" is a very good promoter of robotics.

The parallel reality is the most virtual. There are a lot of forums, groups, and mailing lists about these projects, with many participants, each with his own lab and many ideas to share. The "circuit cellar" is a dream for every one of us, but we have not yet reached the critical mass in any town to organize one.

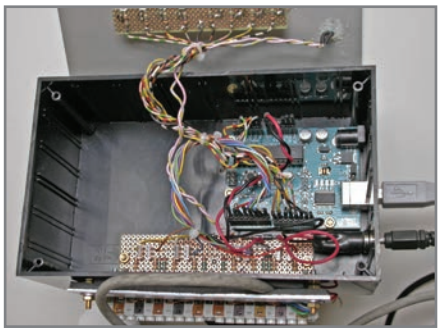
NAN: Robotics is a focal theme in many of the articles you have contributed to *Circuit Cellar*. One of your article series, "Robot Navigation and Control" (224–225, 2009), was about a navigation control subsystem you built for an autonomous differential steering explorer robot. The first part focused on the robotic platform that drives motors and controls an H-bridge. You then described the software development phase of the project. Is the project still in use? Have you made any updates to it?

GUIDO: After I wrote that article series, that project evolved until the beginning of this year. There is a new switched power supply, a new audio sensor, the latest version of dsNav dsPIC33-based board became commercially available online, some mechanical changing, improvements on telemetry console, a lot of modifications in the firmware, and the UMBmark calibration performed successfully.

The goal is reached. That robot was a lab to experiment sensors, solutions, and technologies. Now I'm ready for a further step: outdoors.

NAN: You wrote another robotics-related article in 2010 titled, "A Sensor System for Robotics Applications" (*Circuit Cellar* 236). Here you describe adding senses—sight, hearing, and touch—to a robotics design. Tell us about the design, which is built around an Arduino Diecimila board. How does the board factor into the design?

GUIDO: That was the first time I used an Arduino. I've always used PICs, and I wanted to test this well-known board. In that case, I needed to interface many I^2C, analog sensors, and an



An interface between PIR sensors and an Isadora program

I²C I/O expander. I didn't want to waste time configuring peripherals. All the sensors had 5-V I/O. The computing time constraints were not so strict. Arduino fits perfectly into all of these prerequisites. The learning curve was very fast. There was already a library of every device I've used. There was no need for a voltage level translation between 3.3 and 5 V.

Everything was easy, fast, and cheap. Why not use it for these kinds of projects?

NAN: You designed an audio sensor for a Rino robotic platform ("Sound Tone Detection with a PSoC Part 1 and Part 2," *Circuit Cellar* 256–257, 2011). Why did you design the system? Did you design it for use at work or home? Give us some examples of how you've used the sensor.

GUIDO: I already had a sound board based on classic op-amp ICs. I discovered the PSoC devices in a robotic meeting. At that moment, there was a special offer for the PSoC1 programmer and, incidentally, it was close to my birthday. What a perfect gift from my relatives!

This was another excuse to study a completely different programmable device and add something new to my background. The learning curve was not as easy as the Arduino one. It is really different because... it does a very different job. The new PSoC-based audio board was smaller, simpler, and with many more features than the previous one. The original project was designed to detect a fixed 4-kHz tone, but now it is easy to change the central frequency, the band, and the behavior of the board. This confirms once more, if needed, that nowadays, this kind of professional design is also available to hobbyists.

NAN: The Arduino Project originated in Ivrea, Italy. Have you used it? If so, what do you think?

GUIDO: Yes, I've used it and I use it often. I became an Arduino fan after the aforementioned sensor board project. Before then, I was very skeptical. I love all the openness philosophy and the community behind Arduino. It is an idea out of the ordinary, an international collaboration and… I'm proud it is an Italian idea. It is not so powerful in hardware, but you can easily solve most of your electronics issues with it.

I've read a lot about Arduino on many sites and magazines such as Wired Italia (it was on the first published number).

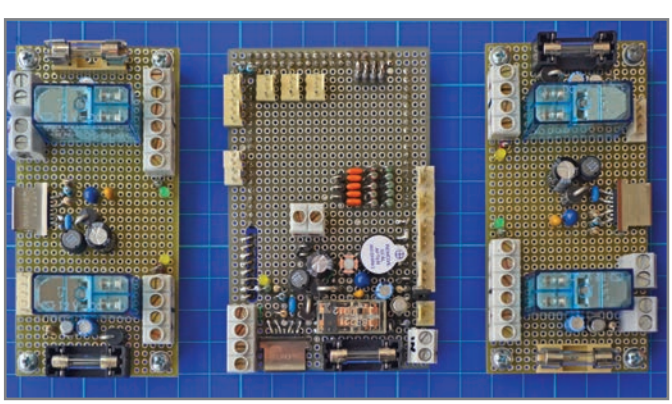
I've also personally met Massimo Banzi at the Romecup 2011. Everything I've read was confirmed by those hours I spent talking with him. He is very friendly and he is an explosion of ideas. In that meeting he fascinated many teachers who started using Arduino too in their lab courses.

I've included several pictures of some of my Arduino-based projects.

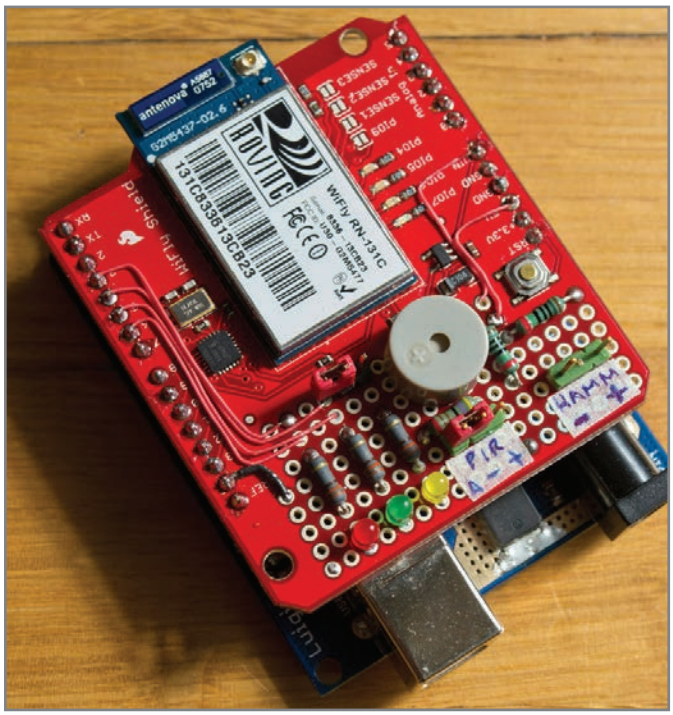
NAN: What are your go-to microcontrollers and embedded platforms these days? Do you have favorites, or do you use a variety of different chips?

GUIDO: I was brought up (in terms of robotics) with Microchip devices. I've used at least one device for almost each PIC family, PIC10F, PIC12F, PIC16F, PIC18F, dsPIC30F, and dsPIC33F. I have an ICD3 and I'm very happy now that the new MPLABX also runs natively on Mac OS X. The flexibility of those devices is very high and you can also develop very sophisticated systems. In one of my latest projects (modifying a parking sensor to obtain a quad sonar) I confirmed once more (if needed) the power of the peripherals also with a "simple" PIC18F2620. Almost the entire project is realized just configuring and connecting peripherals. The software is just simple glue that further connects the peripherals. The exact control you can have on every single bit or execution cycle gives you the opportunity to obtain whatever you want. I use them for all my critical projects, when I need to squeeze the maximum performance by applying all the optimization concepts I know.

On the contrary, if you need to quickly build something very simple, is not so easy with standard Microchip tools. You have no reliable, already-done libraries to configure the peripherals that often differ enough among devices. You spend most of your time just searching for that register's bit that will solve your problem. There is a huge amount of very good documentation but… you have to spend a lot of time.



A homemade shield for an Arduino mega used to create a subsystem that controls batteries and power supply status for Guido's forthcoming robot



Using a WiFly shield to connect an Arduino to the 'Net

In my opinion, this is one of the reasons for the Arduino success. Sometimes we are lazy, or we need short time-to-market (even for a hobbyist), being able to obtain something quick but not dirty, even if not extra-optimized. So, in some more relaxed projects I've used some of the many *xx*-ino boards.

At some point, when I've substituted any of my old friends, SN74*xx* or NE555, with an MCU of whichever type, I wanted to do the same also for my even older analog friends: op-amps. When I need some analog circuit, amplifiers, filters and similar, it is easier to use a PSoC in place of a bunch of transistors, op-amps, capacitors, resistors, and so forth. The modification of a diagram or some line of codes on an IDE is much more friendly than soldering.

I've started studying the amazing world of GPS, in many of its details. I've found some free libraries to do all the not-so-simple calculi for distance, bearing, and so on, but I needed a processor more powerful than a dsPIC to execute, in real time, all that trigonometric math. An mbed gave me the power of an ARM with the easiness of an Arduino, once again with free IDE and compiler.

In conclusion, I always prefer my "first love:" PICs, but there are so many other devices that I can find the one that best fits any project on my mind.

Furthermore, because this is a hobby for me—and I've got a lot of knowledge from many contributors on magazines, websites, and other papers—I like to share back what I've learned. Using different solutions give me the opportunity to do this on the same media: paper, website, or forums. All of my projects are publicly available. Thanks again, Internet!

NAN: Do you have any advice for *Circuit Cellar* readers who are considering building their own autonomous robots?

GUIDO: I love robotics because it involves so many disciplines. Each one of us can find their own specialization, but we must also know something about other fields. I receive a lot of odd questions in the forums coming from people who, knowing just high-level programming, believe they should immediately be able to build an autonomous robot.

Allow me to use the words of one of my gurus, David P. Anderson. Speaking about a robot competition, he said that you must accomplish in advance the very first test: make the robot run forward and backward for some minutes:

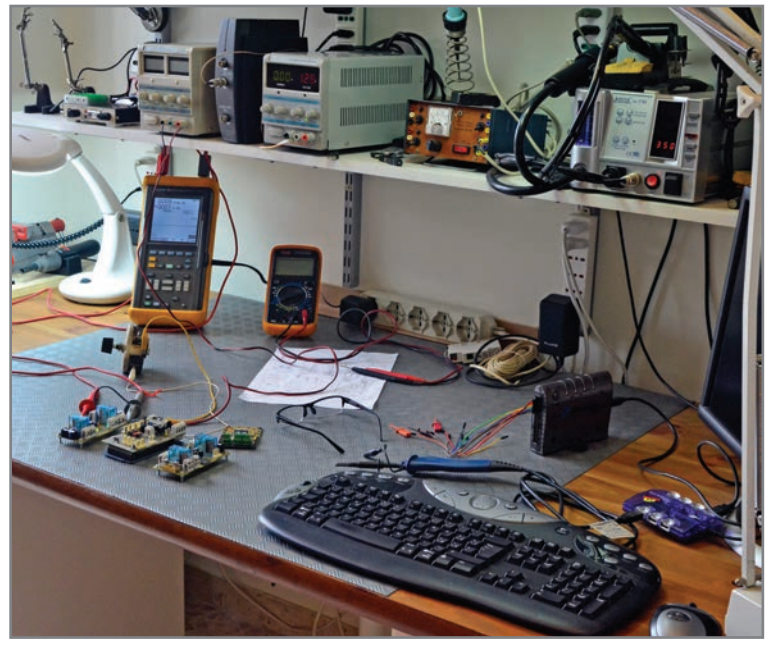
"Seasoned robot builders will tell you that getting to this stage is most of the battle. To successfully run this first simple exercise, the robot builder must have some sort of functioning robot platform with wheels and motors and batteries attached, all mechanically and electrically sound; some sort of H-bridge or other means of controlling the motors from a microcontroller; a microcontroller with the necessary I/O all wired up and working; a software development environment set up and working, with the ability to connect to the robot and download code; and a robust enough implementation that it can run for 10 or 15 minutes without crashing, resetting, coming apart, or having motor EMF spikes brown-out the CPU."

You have only a few ways to do that: One, work in a team with distributed specializations; two, buy a kit with most of those jobs already done; or three, study, practice, study, practice, study and… study again. That's all!

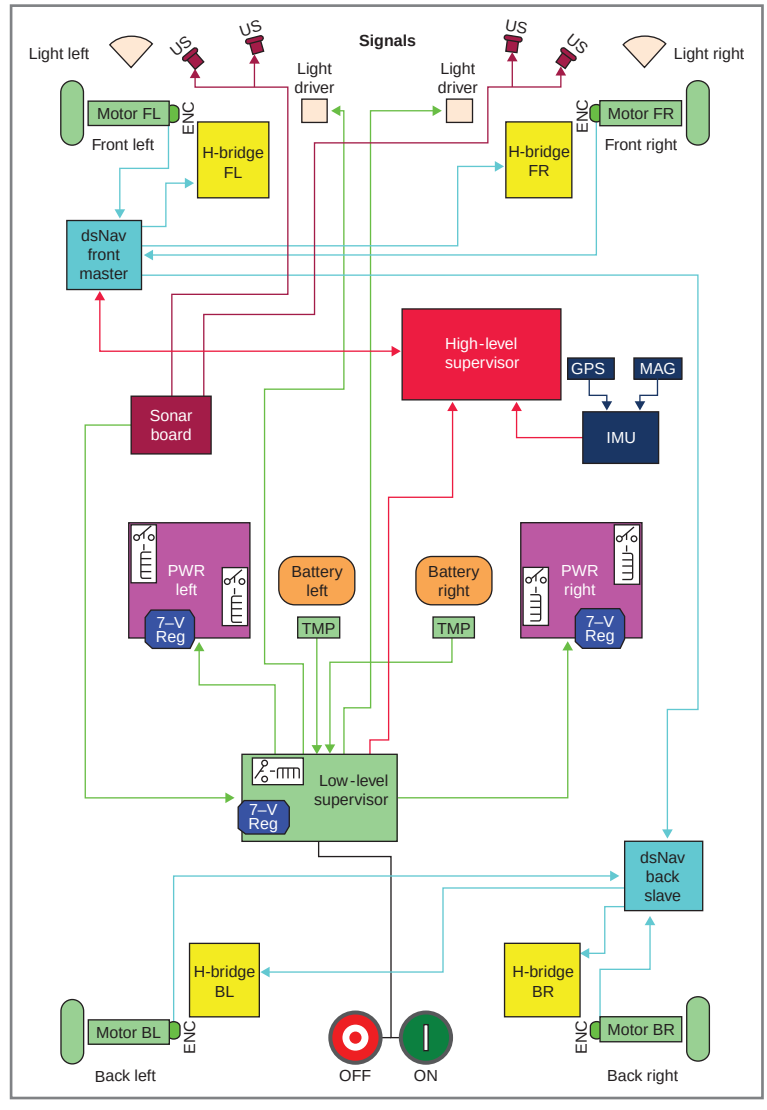
As you can see from my lab picture, I've chosen the third way. I excluded the first way, because this is a hobby performed just in my spare time (that I am not able to schedule), I can only share experiences sometimes and/or remotely with my friends. I excluded the second way because I want to build everything by myself, but this is only a preference of mine. There are a lot of kits that enable you to have a working robot with little or no mechanics effort: LEGO MINDSTORMS and VEX supply building blocks, Parallax and Pololu have already done robotics platforms of many sizes, just to talk about the most famous ones. It must be clear that this one is not the cheapest path and you must know something about electronics anyway.

On my workbench you can see at least three disciplines from left to right: mechanics, electronics, and programming. But they are not enough. You require physics, math, and more. The third way is not the easiest one, but it's the one with the most satisfaction.

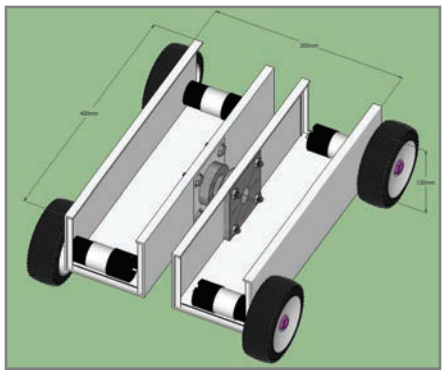
NAN: Are you currently working on or planning any robotics-related projects? Can you tell us about them?



A glimpse at Gudio's workbench reveals at least three disciplines, from left to right: mechanics, electronics, and programming



A block diagram of the boards that will be inside Guido's outdoor robot



The design phase of Guido's Lino robot: a 3-D model created with Google Sketchup

A mockup of the Lino made with corrugated paper to test the validity of the project

An aluminum sheet that has been cut and folded for the final version of Lino

GUIDO: Yes, I am.

Once I wrote in a forum, "I'll never build a robot that cannot fit in a shoebox." This was because my lab is in an apartment, and I have limited mechanical capabilities. Luckily, the human mind can change an idea during its lifetime. After my indoor studies were over, I started thinking about an outdoor robot. This must be simple, cheap enough, and realizable by anyone who is patient enough to follow my explanations. To make a full suspended four-wheel drive robot without using expensive RC cars parts that are not easy to assemble, I was inspired by the award-winning Flexi Flyer, articulated robot.

More pictures are available at my Lino website (www.guiott.com/Lino/Sito/Lino.html).

Right now, I'm in a very early stage. I want to reuse the already acquired knowledge for the "low-level" subsystem (batteries, sensors, and motor control), because the problems are very similar to the indoor experience. After everything works on this part (Anderson docet) I'll start experimenting with something new for me: gyros, accelerometers, magnetometers, and GPS. In short, inertial measurement unit (IMU).

I don't know when it will be ready, but it will be ready. Stay tuned.

NAN: What do you consider to be the "next big thing" in the embedded design industry? Is there a particular technology that you've used or seen that will change the way engineers design in the coming months and years?

GUIDO: As often happens, the "big thing" is one of the smallest ones. Many manufacturers are working on micro-nano-pico watt devices. I've done a little but not very extreme experimenting with my Pendulum project. Using the sleeping features of a simple PIC10F22P with some care, I've maintained the pendulum's oscillation bob for a year with a couple of AAA batteries and it is still oscillating behind me right now.

Because of this kind of MCU, we can start seriously thinking about energy harvesting. We can get energy from light, heat, any kind of RF, movement or whatever, to have a self-powered, autonomous device.

Thanks to smartphones, PDAs, tablets, and other portable devices, the MEMS sensors have become smaller and less expensive.

In my opinion, all this technology—together with supercapacitors, solid-state batteries or similar—will spread many small devices everywhere to monitor everything.