APRIL 1997 ISSUE OF THE DPRG NEWSLETTER

Month after month, we are growing in attentance at the meetings. It seems we've increased about 5 to 10 each month. At the last meeting we hit over 45 in attendance! At this rate, just think of how large the attendance could be next meeting? However, as everyone knows, we can't spell success without u! So, be a statistic and fill a chair at the April meeting! Believe me, you'll be glad you did. Come see what all of the excitement is about!

Hope you have a happy easter (Did anyone make a robot that hunts for eggs?).

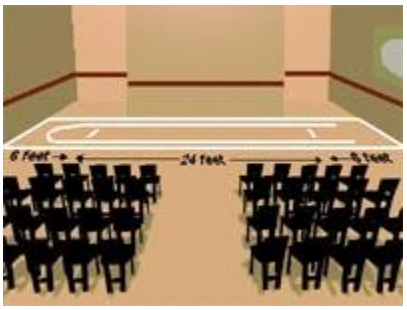
The Dallas Personal Robotics Group is one of the nation's oldest special interest groups dedicated to the development and use of personal robotics and has been around since 1984.

Look at the huge crowd at the March meeting. Offically we had 45, but a few more came in later pushing to total to over 50!!! If you missed this meeting, you really picked a bad time to miss! We had so many great things to give away and so many great robots to see!

DPRG's Upcoming Contest

DPRG will be having a robotics contest in May with trials to start in April. The purpose of the contest is to raise participation in the group. Hopefully this will inspire those who have half-built robots to get them built so that they can do the course. Last time (Nov 93) we had the contest many said they'd have a robot, but only 4 actually tried (although there seemed to be some dispute about that number). Roger Arrick's D-Bot won.

At the February meeting we talked about wall hugging robots could use the side wall of room 1061 with the end wall being the back wall of the room. That would allow a wall along the course and a wall at the end of the course. This is a slightly modified course from the one in the February Newsletter.



The last contest we had was in November of '93. Everyone I spoke to said that we should have contests on a more regular basis (than every 3-4 years). Some mentioned that they'd like to see a contest every 3-4 months.

The simple premise for the contest is for a robot to autonomously go from the start area (on the right) to the end area (on the left), and back in the best time. Although this seems like a simple task, it's amazing how few were able to come up with a robot that could do it for the last contest.

The contest area will be the length of the current room we are meeting in. The length will be roughly 36 feet total: 6 feet on each end and 24 feet in the middle. Also the contest area will be 6 feet wide. The bounds of the contest area will be marked with 1 inch masking tape. For line following robots, a 1 inch piece of masking tape 1 foot from any bounds will run the back side of the course, will have a turning radius (about 4 feet in diameter) end the end area, and will run the front side of the course back to the start area (again 1 foot from the bounds). The tape designating the start of the end area will not intefere with the line following tape 6 inches on either side.

Garp Unveiled

GARP was unveiled at the March meeting. GARP is the DPRG's Group Autonomous Robotics Project. Remarkably, GARP has gone through a drastic transition from the last meeting. It went from just having a non-working wheelchair to what you can see in the picture to the right. The GARProject is really beginning - a dream DPRG has had for several years. We need to really get excited and get GARP going and live our long envisioned dream!

James Vroman of Tech Tools who is also a DPRG member spent very much time working on GARP. So far, he has figured out the wiring schematics, figured out the motor interface (there's gobs of wires coming out of each motor - who knows why), and he has built a wooden case to contain the electronics.

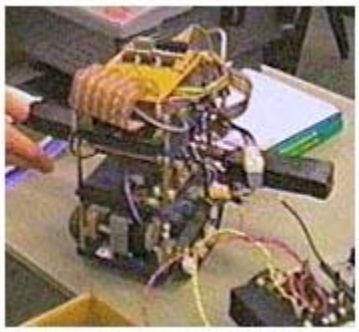
So far very little money has been spent on the GARP project. The wheelchair itself was donated by a guy in New Jersey who worked at a medical supply company. The wheelchair that was donated was not in a working condition, so it was donated to the DPRG. (Within a short time, James Vroman got it working!) The wooden case was built by James with parts found in a dumpster within a 100' of the building in which he works. (I have a feeling that now that everyone knows about this goldmine dumpster, we'll see a lot of DPRG members hanging around the Tech Tools dumpster!). Several other items have been donated for the CPU of GARP. Dan Mathias another DPRG member has donated a hard drive and monitor. Roger Arrick, a DPRG long time member and past DPRG president, has donated a 386 motherboard. Many others have contributed time and input to GARP as well.

During the meeting James Vroman talked about the various aspects of what he has done so far. GARP was made to be very modular - it can be disassembled in about a minute and compressed to fit in a car! James also gave a quick demonstration of the power that GARP has. He flipped a couple of switches on the back of GARP, grabbed the controller with the three foot cable and walked behind it as he sent it roving around the room. GARP currently is wired for two speeds, and at either speed it has plenty of power. Remember the wheel chair was designed to carry about a 200 pound person! It can easily pop a wheelie!

GARP was definately the spotlight of the March meeting. Many were enthusiastic about the GARProject. So many questions were asked about GARP and what we are planning to do with it. The funny thing is that we the members are the ones who decide what will become of GARP. So far we are talking about having a PC be the main brain of GARP running an operating system like Linux. 8051s and HC11s will most likely be used to communicate to the main brain and run perifreals such as arms and sensors.

The aMAZEing ARMS and SPUD

Chad, a visitor to the DPRG meeting brought his robots to show. Chad is a sophmore of University of Texas at Arlingtion and has worked on these robots as part of the Autonomous Vehical Systems of UTA group.



The ARMS robot (or Autonomous Robotic Maze Software) pictured to the left was used to run one of the robotics contests of running a maze autonomously. The ARMS robot is based on the HC11 processor. The wings that extend from the sides and the front are used for navagation and have infra-red sensors pointing down to detect walls. The SPUD is based on a tank track design using large steppers connected directly directly to the shaft.

Also pictured is another robotics project of the same group. This robot is called SPUD (or Systematic Paranoid Unmanned Device). It is based on the 68332 processor which Chad says is the main MCU that the group uses. SPUD untilizes a radio-controlled chassie and uses a servo for steering and a poloroid ultrasonic sensor pointing forward to detect obstacles.

James shows off Linus

James Vroman brought in his Linus Robot that he plans to use for running the DPRG contest in May. His robot is based on a remote control car that he bought at a local department store and wired to his single board computer. The SBC is the Bot-Board from Marvin Green of Nuts & Volts fame and also a member of the Seattle Robotics Society. This HC11 based board is great because it already has outputs for up to four servos and up to four analog to digital inputs. The wire coming out of the back is for the battery power which neatly fits just below the white case which used to be an old tape backup case. The Linus robot also uses a regular servo for steering, and will have infra-red sensors at the front and facing down to detect the masking tape that will be used in the contest. Jamie stated that he hopes to make the software smart enough to detect either a light tape on dark background or dark tape on light background and to be able to easily transition between the two. He says that his sensors will be hooked into the analog inputs of his SBC, so he will actually be able to read levels of illumination so he claims to have a much higher level of reliablity. He said that his boss already agreed to allow him to run tape along the floor at his place of employment so that he can do trial runs at work! It looks like James' Linus robot will be quite a contender for the contest in May!

Jamie shows us his leg

Jamie Merrill another Tech Tools employee brought his walker robot in it's unfinished state to show the work he has completed so far. He says he's still trying to come up with a name for his robot. So far he plans to use six Hitec servos for the entire robot. Two servos for each leg, and one servo each to operate the hips. On each leg there is one servo for foward and backward swing, and another servo to operate the knees and linkage via pulleys to operate the ankles. Jamie's robot is very frugally made with parts he has dug up around the shop. The bearings he is using comes from an old Commodore computer's floppy drive hub. The lower leg is made from an IC tube, and the upper leg is hacksawed metal from a computer's case. The foot is made of an old Atari mouse. He plans to use the buttons on the mouse as sensors for various things. So far it really looks like an ingenious design. Jamie hopes to have his walker robot run the contest in May.

Quadravox gives a presentation of their speech product

Quadravox is a new company that specializes in PC based speech development systems. Their aim is to deliver affordable, user friendly, Windows compatible software and hardware packages that for the first time make it easy to add a speech (or, more generally, sound) interface to any stand alone system or product.

At the March DPRG meeting Quadravox gave a brief presentation of their products. The Qbox system utilizes a PC's sound card to decrease the cost of a development system. Most PC's have a 16 bit sound card capable of recording from a microphone, so instead of developing a complete recording device, Quadravox will just accept common WAV files recorded from any sound card.

The development system (the QVT-11) is a programmer device that burns WAV files onto a special sound chip. The programmer can be hooked up to your parallel port via a parallel printer cable. Using software that Quadravox provides you can do the common device programming functions as you would an eprom such as blank check, and program/verify, etc. The special chips will hold up to 60 seconds of speech or sound from WAV files recorded at 11 khz.

After you have burned one of their special chips with the WAV file of your choosing, you plug the chip into one of their serial output modules. The serial output module seems to allow control via a serial port, a keypad, a parallel port, or X10. The output module takes a simple 9v battery to power it. On board it comes with a LM386 amplifier chip, so you should be able to hear the speech by simply hooking up a regular bare 8 ohm 2 inch speaker. The output modules are really small at about 2"x2.5" and have a socket for the 16 pin speech chip, a 9v battery connector, and a 1/8" speaker port, and a interface port (for serial or parallel connection). Control via a serial

port will allow 128 sequences and 31 individual module addresses plus remote control of device output lines for user functions!

The iQbox speech development system, pictured at left was demoed at the March DPRG meeting. The iQbox is their more impressive version of the programmer and will be able to handle longer duration future chip versions. There's so much that these devices can do, that there's no way I could fit all of the features in this short article. Suffice it to say that it looks like you can do just about anything with the speech you have recorded from concatenating words to picking particular words for output.

The software that comes with the system seems to do many things. It can compress the speech using some linear predictive coding algorithms, and also allows sound editing.

The prices for the Qbox system is $90 for the programmer, $15 for the output module, $10 for the special chip, also you'll need a wall adapter for the programmer at about $10. Be sure to check out their web page for special prices, and prices for packages, you can get it all for a little cheaper. For the iQbox system the development system is $260.

Quadravox was so kind to donate two of their Qbox basic kits, two power supplies for the kits, 2 assembled serial output modules, and 3 of the TI's One-Time-Programmable Speech Synthesizer chips. I'll try to play with it soon, and we'll hopefully have a review of their products. All in all, it looks like an easy and inexpensive way to have professional quality speech or sound on any product.

Hopefully, we can have our GARP project talking without any problem!

Robotics Digest now in Print

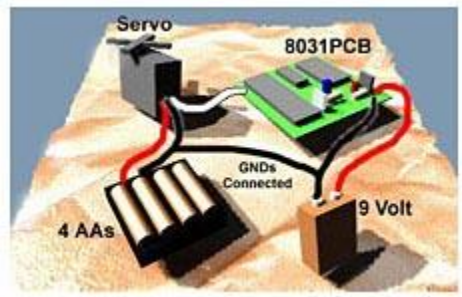
The new Robotics Digest magazine is now in print. This magazine is the predecessor to the Robotics Practitioner magazine that folded in the Summer of 1996. William Gates, the sole publisher of the new magazine believes that due to his somewhat lower costs in producing magazines, he can succeed where others have failed in this tight niche market. I for one hope that it does succeed, and from first glance at the magazine, I'm very impressed at what a one man publishing company can do!

Turning the to table of contents of the new magazine was like turning to the hall of fame on the comp.robotics.misc newsgroup. We even have a couple of members from our group with by-lines in the magazine. The magazine has great articles, great advertisers, you name it, it's great. If you're into robotics, and your not subscribed, you're really missing out. The magazine is $24 a year sent out quarterly (4 issues). Here's the address: Robotics Digest, William E. Gates, 966 Elm St, Rocky Ford, CO 81067, phone: 719-254-4558 or fax: 719-254-4517.

Steve Rainwater, a DPRG member, contacted the publisher of the new Robotics Digest magazine. In doing so, Steve managed to get the publisher to send us 45 magazines for samples that we could hand out at the meeting. If you missed getting a magazine at the March meeting, Tyce Elkins (the DPRG's librarian) may still have a few. This is the charter issue. It may really be worth something some day.

My Experience with Servos

by Jim Brown



I don't claim to be an expert on servos, but I was able to get a servo to

work so I thought it might be nice to tell my experience with servos in the hopes that it might help someone out there. Disclaimer: However, if you fry something after reading this, don't come after me, you should do your homework better.

I sort of started hap-hazzardly. I decided I wanted to make a robot with servos. I had no design yet, nor a servo controller design or software yet, I just blindly purchased 3 of the cheapest servos from Tower Hobbies (http://www.towerhobbies.com) with the hopes that I could get them working. While I waited for them to return I began to work on the software and hardware design. In a matter of just a couple of days Tower Hobbies shipped my servos. They were very fast about it. I hadn't even had time to work on a design.

With just what I thought I knew about servos, I wrote some code in Keil C, but I had no way to test it. Then, I started by asking a few people what they knew about servos, and I also searched around on the internet for some information on servos. It's amazing what you can find with an internet search engine!

For power, I found out that most servos operate on 4.8 volts, but are ok with anything from 4.0 volts to 6.0 volts. Servos have three wires, and usually are: Red=+, Black=Gnd, White=Signal. The signal line should be at TTL level.

For the signal that controls the positioning of the servo, it needs a pulse of 1ms to 2ms and that pulse needs to be sent every 20ms to 30ms. The positioning goes that a pulse of 1.5ms is the shaft center position, and any other pulse width would move the shaft proportionally to the left or right.

For a proto design, I used one Tower Hobbies TS-53 servo, four AA batteries for the servo power, my 8031PCB for the brains, and a 9V to power the 8031PCB. I connected the grounds so that the signal line should have a path back to the 9V. Having seperate power for motor and SBC keeps unwanted noise from the motor from affecting the CPU (or so I'm told).

After reading some info about servos on the internet, I rewrote my servo controller code, and tried it out. Viola! On my first attempt, my servo did the test pattern. I expected to have to revamp my code several times before it would work, but hey, if it works the first time, who am I to argue! Click here if you want to see the C source code and hex output file to the servo controller.

The software just goes through a test pattern of: 1000us, 1250us, 1500us, 1750us, 2000us, 1750us, 1500us, 1250us, 1000us, 2000us, and repeats. In otherwords, it goes from left to right in steps, and then right to left in steps, and then quickly to the left and back to the right.

The servo controller software as it is should be able to control up to 8 servos. Both timers of the 8031 are being used, so if one were needing serial control, you'd want to switch to the 8032 (or 8752) version to get an extra timer to handle the serial control. That could make it an inexpensive serial servo controller perifreal device.

My hopes are to make some sort of walker robot with this design.

CUPL Review

by Kevin Carter

My task has been to review the copy of the CUPL package that was purchased by the DPRG for use with the PROMAX device programmer. First of all, this particular package was selected for one specific reason, unlike the shareware and very low cost packages, this one is capable of generating the fuse maps for EPLD's (erasable programmable logic devices) where others only do the standard PLD's, which the PROMAX can not program. Along with the erasability of the EPLD, there are two other advantages to their use: they use less power, especially nice in a battery powered robot, and usually offer lower propagation delays than the standard PLD's. Using CUPL actually turned out much easier than I expected it to be.

The first step, after installing the thing, is to copy the supplied template file to a new file, using the name of whatever your target device is going to be, in the case of my test device, "GLUE", as it is intended to replace the glue logic in the designing of a brain board for a robot I hope to make, probably after I get off of 7 day work weeks (which will be sometime after the Pope's divorce (grin)). The template file is exactly what the name suggests, the template that will be used to design and test the fuse map for the PLD. You start by filling in the header information, as described in the manual, and then designating what the input and output terms will be. The next step is where the real fun (and fustration) begins, setting up the logic equations that determine just what fuses get blown to form the PLD's internal logic. Most of it is very straight forward, simply setting up the boolean equations. What got fun was figuring out the select logic, especially as the monster only has 4 logic functions, not, and, or, and exclusive or. As I found out to my annoyance, you have to be VERY careful how you arrange the logic functions, or you get some very odd results (very large grin). The next step is to set up the logic simulator file.

The logic simulator file starts off with a trimmed version of the template file, as described by the manual, and is then expanded to show the expected outputs of the various logic equations depending on the input conditions. After you get this set up, it is time to actually run the program.

To run a logic compile or simulation, you simply select the run function from the menu bar, give it the information it requests, such as the target device and type of output files that you want, and let it go. I have a suggestion, and a comment. First, do not run the simulator from the compile routine, as it takes considerably longer, and a minor bug in the simulator program can cause your device compile to fail. Second, the simulator program seems to have a slight bug, as it tries to run a part of the simulation that does not appear to be there, you will see what I mean when you try it, but this does not appear to affect the simulation output. A compile run generates several different outputs, first the fuse map that is used to program the device and is also used to generate a simulator output, a list file that is very useful in finding typo's (I hate that part!), and a documentation file that tells you what is going on. Running a logic simulation generates a simulator output report that lets you know if (and when) you get it right.

Once you get the logic right, it is a simple matter of using the Promax to program the device, and you are up and running. It can really make a major difference in your schematic, it reduced my chip count from 19 to only 10 devices! Good luck, and have fun!

Editor's note: Kevin was so kind to also include the sample source for his devices spoken of in the article. I hope to have them typed in and online soon.

Roger Shows his 1bit Robot

Roger Arrick brought in his 1 bit robot that he previously ran in the contest back in 1993. He made the 1 bit robot just to show how easy it could be to make a robot to be in the contest. At the March meeting he said that he would not enter that robot into the contest in May. Maybe if we bug him enough we could persuade him to demo the robot in the contest in May. The 1 bit robot circuit is just a DPDT switch attached to a dowel rod. It goes across the room as fast as it can, when the dowel rod hits the wall it will cause the switch to switch causing the motors to go into reverse. The 1 bit robot is made of a two-by-four, a toy car part for the rear axel a large front wheel, and the dowel rod and switch at the top. There you have it a cheap simple autonomous robot that can run the contest. Maybe seeing Roger's creation will inspire all of us members to make at least something to run the contest.

Keil Software gives away demo kits

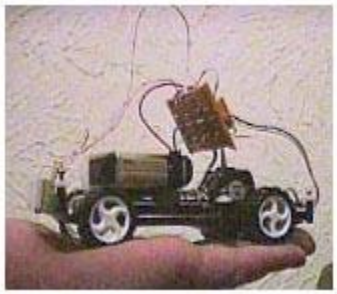
Keil Software graciously donated 18 of their CD-ROM demo kits of their 8051 C compilers. This donation was in addition to all of the previous donations of the 3 8051 full versions of their C compilers and 2 of their evaluation boards. At the March meeting these demo kits were quickly grabbed up. The demo kit is just like the full versions except they will only compile code up to 2k. If you didn't get a demo kit, probably if you call up Keil Software and ask really nicely, they'll probably send you one. Keil Software's C compiler is one of the best in the industry for compiling 8051 C code in a highly optimized fashion for embedded SBC's. Many thanks to Keil Software's generous donations.

Tech tools gives away 8051 boards

Tech Tools graciously donated 10 of a prototype 8051 board. These board had a simple problem which made them unusable for production, but for hackers like us that don't mind a simple fix, it was no problem. These boards were quickly grabbed up at the March meeting. We all agreed at the meeting that these would be givaways for new members. We actually only received 3 new members at the March meeting, but we allowed some others to have them as well. The board's problem is a simple swapping of the 7805 chip's pins. The simple modification is to solder the heat sink to the board's ground plane, clip the ground pin, and bend the 12v pin to the center hole. Hopefully those that received a board will use them for their robotics projects and possibly even enter such a robot in the contest in May. And, Tech Tools gave away several pocket clip flat head screwdrivers. Big thanks to Tech Tools for their many donations.

Parallax send Catalogs

James Vroman contacted Parallax recently and managed to talk them into sending us a box full of their catalogs. These catalogs were quickly picked up at the March meeting. Parallax has many great PIC and basic stamp type products.



New Electronics Catalogs from Future Active

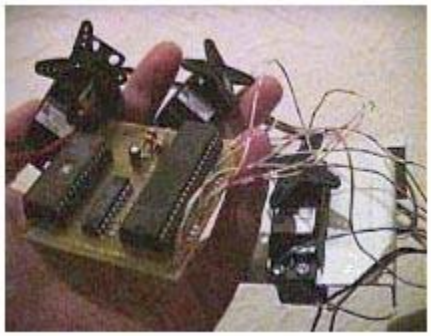
Future Active, a new electronics supply company sent James Vroman some of their catalogs to hand out at the March meeting. They have really nice full color catalogs and they seem to be in competition with Digikey. From looking at their list of parts, they seem to have everything a robotics enthusiast could want from really hard to find ICs to the regular tools and stuff. It is definately worth checking out. I couldn't seem to find a web page for them, so I'll list their address: Future Active Industrial Electronics, 800 East Campbell Rd. Suite 126, Richardson, TX 75081, (214) 231-7195 or (800) 272-0694 or Fax: (214) 231-2508.

Jim Shows his Robots

I brought my three robots that I hope to enter into the contest in May.

The first was my 2 bit robot. It is made out of two latching relays and on a simple toy car base. It has a button the front and a button on the back. It's just a little more powerful than Roger's one bit robot above.

My Baby2 robot is based on my 8031PCB. It uses a RC car base I found in a thrift shop, and ties directly into the car's h-bridge that was already on board (I just removed the RC receiver stuff). Right now it just goes through a test pattern of just Pulse Width Modulating the motors to spin the wheels at different speeds.



The third robot I brought was my Baby3 robot. It's the least likely to be finished by the contest. So far it's just 3 servos and my 8031PCB. So far the servos go through a test pattern, but I do not have a walking robot yet. My hopes are to make a hexapod walking robots using the electronics shown. Maybe there'll be more to show by the next meeting.